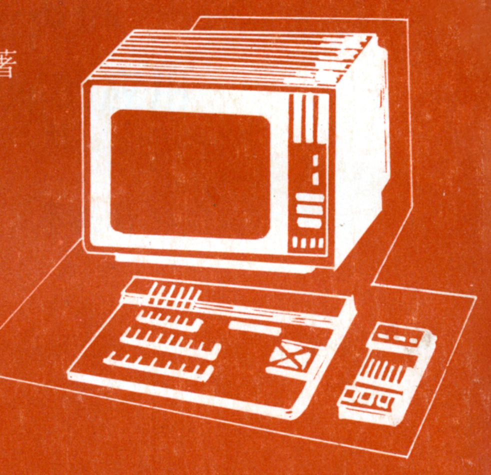
中华学习机

韩仲清 廖兴祥 编著



汉字
数据库的应用

成都三开元电脑部

中华学习机
汉字数据库的应用

韩仲清　　廖兴祥

成都三开元电脑部

1989.7



前　　言

中华学习机以前所未有的速度进入家庭，并在开发青少年智力方面已经显示出了威力。电脑时代的到来，比我们想象的还要快。

这是继《如何使用中华学习机》之后，我们献给孩子们的又一本电脑书籍。希望她能给广大电脑工作者、家长、特别是青少年带来知识，带来智慧，带来力量。

本书按教材形式编写，书中给出了大量的实例，各章附有习题。这些为读者尽快掌握 *cdDBAE*Ⅱ的知识提供了方便。

本书在编写过程中，得到成都三开元电脑部舒新生同志的大力支持，在此表示感谢！

书中不妥之处，请读者指正。

编者

1989.7.10

于四川大学计算机科学系

内容提要

本书详细介绍了中华学习机的汉字数据库系统 *cdBASE* Ⅱ的使用方法和技巧。内容包括:

*cdDBAE*Ⅱ的基本概念；中华学习机上如何安装和运行 *cdBASE*Ⅱ；数据库文件的建立和操作方法；函数和设置运行特征；排序和索引；命令文件和报表格式设计；数据库应用实例；*cdBASE*Ⅱ程序设计的方法与技巧；支持数据库的 *CP / M* 操作系统介绍。

全书内容充实、具体、实用、通俗、易懂。可作职业学校、中小学、中专、非计算机专业学生数据库课程的教材，也可供从事计算机研究和应用的人员参考。

本书一至六章由韩仲清编写，七至九章由廖兴祥编写。韩仲清统稿。

本书由四川省地质矿产局计算中心电脑排版

目　　录

第一章　cdBASEⅡ的基本概念

数据库技术的应用已经渗透到国民经济的各个部门。然而，在一般人的心目中，认为数据库似乎只有在大容量的机器上才能使用。的确，最初的数据库是建立在大型机的基础之上的。现在，象中华学习机这类微型机，照样也可以使用数据库，这就是本章将要开始介绍的汉字 dBASEⅡ关系数据库。

1.1 什么是数据库？

提到数据库（DataBasc，缩写为 DB）这个术语，可能读者并不陌生，因为很多广告、新闻报道中经常提到。然而，要确切而严格的说出“什么是数据库”又是困难的。因为它不象数学公式和物理定律，三言两语就能说明白。我们从以下几方面来说说“什么是数据库”。

1.1.1 数据库的特点

由于要正面回答“什么是数据库”这个问题比较困难，因而很多资料都采用先“谈谈数据库应该具备一些什么特点”的

办法来讨论数据库。我们也可如法炮制。归纳起来说，数据库有如下一些特点：

1.数据的结构化

所谓数据的结构化，是指数据库能够描述的不单是某一个独立的、零散的事物，而且还可以表达一个事物和其它事物之间的联系。例如：学生是一个事物，刻划学生的特征可用：学号、姓名、性别、年龄等等。数据库当然可以描述学号、姓名、性别、年龄之间的联系。这种联系称为事物的内部联系。课程又是另一个事物，其特征可以是：课号、课名、学时数、先修课程、后继课程等等。同样，课号、课名、学时数、先修课、后继课程刻划了"课程"这一事物的内部联系。对这一内部联系，数据库也能表达。学生和课程虽然是性质不同的两种事物，而事实上它们之间却有一种天然的联系，这种联系就是：一个学生可以选修若干课程；一个课程可有若干学生选修。学生和课程之间存在一种称为"学生选课"的联系，这种一个事物和其它事物之间的联系称为外部联系。数据库技术除了能够描述事物的内部联系以外，还能够描述如象"学生选课"这类外部联系。目前流行的高级语言（如：BASIC、FORTRAN、COBOL 等）的文件系统，通常只能描述事物的内部联系。数据库与高级语言文件系统的根本区别也就在于此。

学生和课程的数据，经过结构化抽象以后，如下图所示：

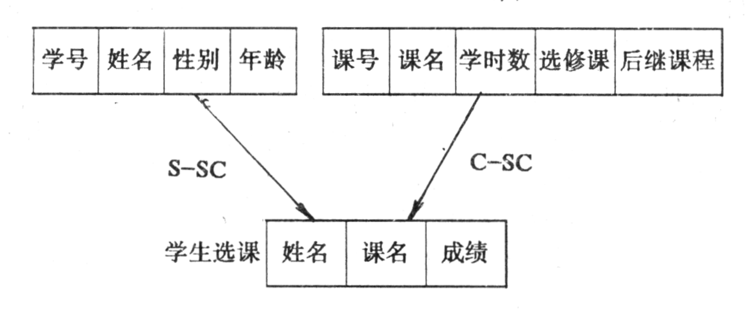


图 *1.1* 数据结构化示意图

这就说明，如果我们要用数据库来解决某单位（工厂、机关、学校、厂矿或医院等）的数据处理问题（如工资、人事、产品等），那么首先必须抽象出单位的数据结构，这种结构也称为企业的整体逻辑结构。

从上可以看出，数据结构实际是企业中同类数据的一个框架。通过数据的结构化处理就把各个框架联系在一起了。

2.数据的独立性

数据的独立性是指数据库中的数据不依赖于具体的应用程序。数据和应用程序分家，这是数据库技术的又一个特点。其好处在于：如果应用程序改变了，存放的数据仍然可以不变；如果数据改变了，应用程序也可以不变。通常，高级语言中程序和数据是紧密相联的，数据为某种应用而存在，程序为某些数据而存在，彼此不能分开。这种方式带来的缺点是显然的。

3.数据的共享性

所谓数据的共享性是指：凡是数据库中存放的数据，从原则上来说，各个用户都可以采用各自的语言和方式来使用

它。例如：学生的数据，除供选课使用外，还可供与之有关的教师、教务部门、后勤部门等使用。彼此的使用可以互不干扰。但需注意，共享决不是随心所欲的乱用，而必须遵循一定的原则。

4.数据的安全性、完整性和并发控制

由于数据的共享性，相应地带来了数据的安全性和并发控制等问题。

安全性是指保证数据库中数据的安全，防止未经授权的人存取或破坏或泄露。为此，采取了一些安全措施。如：规定口令，使用密码、限制存取权限等。

完整性就是要保证数据库中数据始终是正确的。数据库技术采取了一些校验措施来保证数据的完整性。

并发控制对各个用户存取数据的优先顺序和使用情况提供一种协调措施，以保证各个用户存取数据的正确和避免发生冲突。这是由数据库的并发机制来实现的。

5.最小冗余度

所谓最小冗余度，就是指数据中重复数据最少。从理论上来说，可以做到完全没有重复数据，但出于对效率的考虑，允许少量重复。即是说，数据库中数据的冗余是有控制的。

1. 1.2 数据库的种类

目前比较流行的数据库有三种：一种是层次模型的；二种是网状模型的；第三种是关系模型的。三种模型中应用最广、最方便的是关系模型。

1.层次模型

先看图 1.2 所示的层次关系。

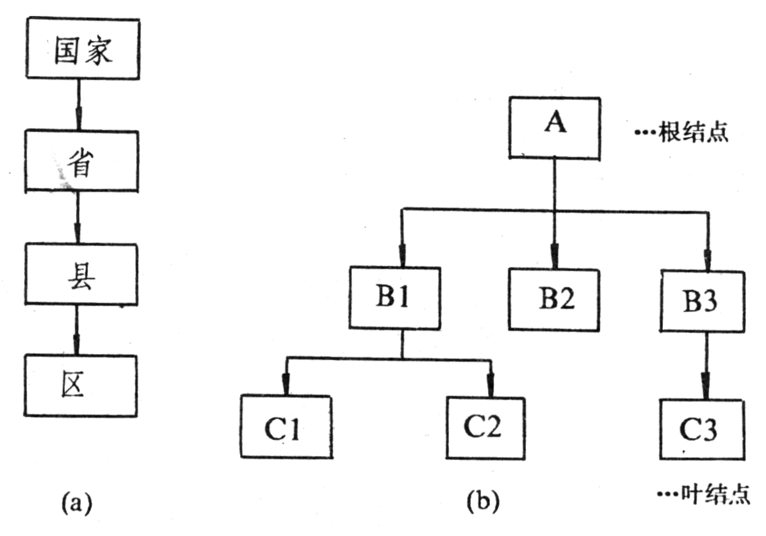


图 *1.2* 层次模型示意图

在（a）中，一个国家下属省，省下属县，县下属区，这是一级管一级，典型的层次结构。在（b）中，A 下属 B_1、B_2、B_3，B_1 又分为 C_1、C_2，B_3 又分为 C_3，这也是一种层次结构。我们将每一个方框称为一个结点。最上面的结点称为根结点。不能再分枝的结点称为叶结点。从上图看出，层次模型符合如下两条：

①只有一个结点无双亲（结点之上不再有管辖的结点，如 A）。

②其它结点有且仅有一个双亲。

由于层次模型类似自然界中一棵倒立的树，因而也称为树形结构。由此，我们不难理解根结点和叶结点的含义。有

的也把箭头上方的结点称为双亲结点，下方的结点称为子女结点。层次模型也类似家族结构。

比较典型而又有代表性的层次模型的数据库是 IMS (Information Management System) 系统。它是 1969 年由美国 IBM 公司推出的。

*2.*网状模型

我们先看图 1.3。

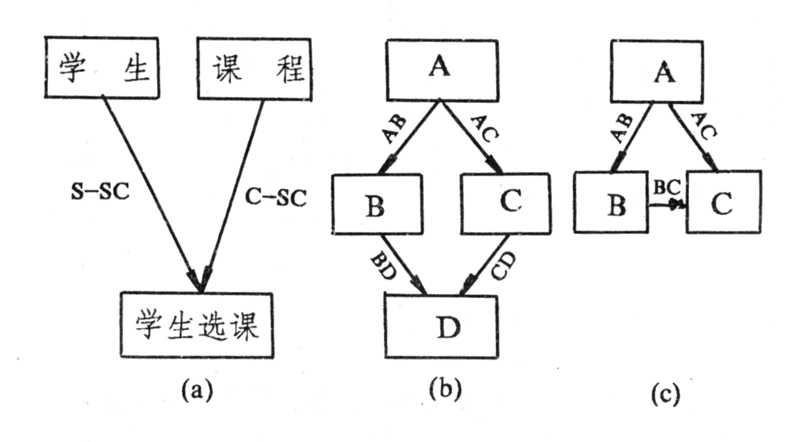


图 *1.3* 网状模型示意图

仔细观察图 1.3 就会发现结点的组成有以下规律:

①可以有多个结点无双亲。(如图 1.3(a))

②至少有一个结点有两个和两个以上的双亲。(如：学生选课结点有学生和课程两个双亲)

满足以上条件的模型称为网状模型。在这种模型中，箭头是极为重要的，它代表了存取数据的方向，称为存取路径。

网状模型的系统研究始于 1969 年美国数据系统语言协会下属的数据库任务组（Data Base Task Group）提出网状模型所应该具备的功能和特征。这个报告几经修改，它对数据库，特别是网状模型的数据库的研究与发展起到了很好的促进作用。以后，人们又在 DBTG 报告的基础上研制了一系列的数据库系统。

3.关系模型

层次模型和网状模型都是在文件管理系统的基础上发展起来的，因而其操作带有浓厚的文件系统的色彩。故而有些资料又把这两种模型统称为格式化模型或叫做传统的格式化模型。虽然格式化模型比起文件系统来说有了很大的进步，但对非计算机专业人员来说仍然是复杂的，这就妨碍了数据库技术的广泛普及和应用。

1970 年，美国 IBM 公司的研究员 E.F.Codd 提出了关系数据库的概念，从而奠定了数据库研究和实践的新方向。

在关系模型中，所有的结点（事物）都用二维表格的形式给出。如前面图 1.1 的学生和课程模型，用关系表示如下：

学生

学号	姓名	性别	年龄
89123	张三	男	15
89112	李四	女	14
…	…	…	

课程

课号	课名	学时数	先修课	后继课程
89001	三角	80	代数	立体几何
89005	数据库	72	BASIC	专家系统
...	...	...	...	...

学生选课

姓名	课名	成绩
张三	数据库	90
李四	代数	85
...	...	...

不论一个企业的逻辑结构有多么复杂，用这种表格总是可以描述出来的。上面的表格有如下一些特点：

①表格的名字——数据库文件名

表格的名字是整个表格的一个代表。如："学生"，"课程"，"学生选课"。这个名字是用户自已取定的。一张表格又称为一个关系。在 dBASE 数据库中，表格名又称数据库文件名。

②表格的栏目——字段名

表格中最上面的一栏是栏目。如：姓名、性别，年龄等是"学生"表格的栏目，它共有四个栏目，在 dBASE 中，表格的栏目称为字段名。如："姓名"是一个字段名，"性别"又是另一个字段名。在关系数据库中，栏目或字段名又称为属

性。一个表格可以由多个栏目（字段）组成。

③表格的内容——记录

表格中有若干横行，每一个横行叫做一个记录。如："89123 张三 男 15"就是"学生"表格的一个记录。在关系数据库中，一个横行又称为一个元组。一般说来，一个表格中可以有若干个记录。

在以后的叙述中，我们把表格和数据库文件、栏目和字段不加区别的等同使用。

自 1970 年以来，关系数据库的研究与应用发展十分迅速。最著名的关系数据库是 SystemR，它是为大型机研制的一个关系数据库，其性能与网状模型不相上下。此外，适宜于微型机上使用的关系数据库有 dBASEⅡ／Ⅲ、FoxBASE$^+$等。目前国内应用最为广泛的数据库是 dBASE Ⅱ／Ⅲ和 FoxBASE$^+$，这两种数据库在用法上大同小异，其命令格式基本上是相同的。因此，只要掌握了这两种中的任一种，另一种也就会用了。

1.1.3 数据库的组成

"数据库"这个词前面已用到多次，但严格说来，可能是不恰当的，因为"数据库"还包含有很多内容。为了全面了解数据库，我们还要引入以下几个概念：

1.数据库系统

所谓数据库系统是指一个计算机系统中配备数据库以后所需人员、硬件和软件的全体。

数据库系统（Data Base System，缩写为 DBS）涉及的人员有两类：一是设计和维护数据库的骨干人员，称为数据库管理员（Data Base Administrator，缩写为

DBA)。对于大型机器来说，通常配备 DBA（微机通常不需要）。要求 DBA 有比较全面的知识，既不但精通计算机，还得熟悉企业管理，DBA 常常由一个专家小组来担任。其职责是:

①建立数据库并负责装入数据。

②监督和控制数据库的使用、运行。

③维护和改进数据库。

另一类人员是数据库的用户。这通常是企业的领导人和工作人员。我们在设计一个具体的数据库时，必须考虑到用户的技术水平。

数据库需要一定的硬件系统的支持。由于各个系统不同，所需硬件要求也不尽相同。总起来说，内外存容量大、机器运行速度快对建立数据库大有好处。

数据库系统涉及到的软件有操作系统(Operating System，缩写为 OS)、数据库管理系统（Data Base Management System，缩写为 DBMS）和应用程序。操作系统用于管理、调度和控制整个计算机系统（包括硬件和软件）。没有操作系统的支持，DBMS 不能正常运行。因而在使用数据库之前，一定要先装入操作系统。应用程序是用户用于检索、插入、删除和修改数据库中数据的一组软件。

2.数据库管理系统

DBMS 是一组非常复杂的软件，它是用户和机器之间的一个“翻译”，它的主要功能是:

①建立数据库

②管理数据库

③维护数据库

④数据通讯

DBMS四种功能都是通过高级语言的形式提供给用户或DBA的。语言分为两类：一是专门用于建立数据库的语言。称为数据定义语言（Date Description Language，缩写为DDL）；另一类是专门用于检索数据库的语言，称为数据操纵语言（Data Maniputation Language，缩写为DML）。

3.数据库

按“数据库”本来的含义，它是指企业中有联系的数据的集合。这些数据通常保存在计算机的外存（如磁盘）上。我们可以用DBMS提供的方法和规则来存取。

数据库、数据库系统、数据库管理系统是互有联系而又有区别的三个概念。严格地说，数据库应当是指“数据”；数据库系统是人员、软硬件的全体；数据库管理系统是用于管理“数据”的一组软件。我们通常所说的学习数据库，实际是学习数据库管理系统的各种语句和语法规则，从而用于设计企业所需要的数据库。

数据库的定义是各种各样的。

J.D.Uuman指出：数据库是存贮在计算机系统中的数据。

C.J.Date说：数据库只不过是一个计算机化了的薄记系统。

J.Martin说得比较具体：数据库是以有控制的冗余度存贮在一起的、内部有联系的数据集合，它以最佳方式为一个或多个应用服务：数据的存贮与应用程序无关；在数据库中增添新数据、修改和检索现有数据用的是一种公共的、有控制的方法。

苏岳威（美籍华人）给数据库下了这样的定义：数据库是从语义上和结构上与任何企业的用户相关联的、且本身又存在相互联系的操作数据的集合。

凡此种种，不一而举。归纳起来，我们可以这样来理解数据库：数据库既表示企业中数据的集合，又表示它还能对数据进行组织、存贮和管理，并提供一套专门的技术和方法，使人们能够简便而迅速地检索、插入、删除和修改库中数据。

如果对这些概念还不清楚的话，可以先学习使用dBASEⅡ，获得一些感性知识后再来温故。

1.2 dBASEⅡ文件概述

1981年5月，ASHTON-TATE公司研制成功dBASEⅡ，以后又经多次修改和扩充，形成了dBASEⅡ的多种版本。由于dBASEⅡ的简便和实用，受到广大用户的普遍欢迎，被誉为大众数据库，1984年7月，ASHTON-TATE公司在征求广大用户意见的基础上，对dBASEⅡ进行了更大的改进，推出了功能更强的dBASEⅢ。目前dBASE的最高版本是dBASEⅣ。

国内使用的dBASEⅡ有纯西文和中西文两种。一般把前者称为dBASEⅡ，后者称为cdBASEⅡ。但两种dBASEⅡ功能相似，用法相同。

dBASEⅡ处理的对象也只能是表格。表格的名字称为数据库文件名，栏目的名字称为字段名，表格的内容叫做记录。一般来说，一个数据库文件有一个名字、若干个字段和

若干个记录。

1.2.1 文件种类及其扩展名

dBASEⅡ提供了 7 类文件:

文件种类	扩展名
数据库文件	·DBF
索引文件	·NDX
存贮文件	·MEM
报表格式文件	·FRM
命令文件	·PRG(·CMD)
格式文件	·FMT
文本输出文件	·TXT

每个文件必须有一个文件名。文件名是字母或数字的序列，字符个数不能超过八个，字符中可以嵌入冒号。注意:①构成文件名的字符只能是字母或数字，第一个字符必须是字母（英文)，后面可以是字母，也可以是数字，也可以直接用汉字作文件名，但不能超过 4 个汉字，因为一个汉字要占两个英文字符位置。②文件名的各字符之间不能夹杂空格(白)。为区别各类文件，对不同类的文件给出了不同的内部标志，这种标志称为扩展名。

数据库文件是 dBASEⅡ中最基本的文件，它用于存贮数据库的基本信息。索引文件是在数据库文件的基础上建立起来的一个有序文件，其作用是便于对数据库文件进行快速查找。存贮文件是把内存变量组织成一个文件，以便永久性地保存内存变量的值。报表格式文件包含了打印报表所需的格式信息。命令文件允许把若干命令组织在一起，从而生成一个文件或程序。格式文件是为了建立常规屏幕格式以便数

据的使用和输出。文本文件可用于系统通讯和记载 dBASE Ⅱ的处理活动。

我们介绍 dBASEⅡ，实际上就是介绍这 7 种文件的建立方式以及与之有关的各种操作技巧。如果能够熟练地掌握这 7 种文件的建立、打开、操作和关闭命令，那么在实际应用中就会更加得心应手，要编写出质量较高的应用程序就指日可待了。

1.2.2 使用符号约定

为叙述的方便，我们约定以下符号的含义。

{…} 花括弧中的内容必选一项，亦可选多项。

[…] 方括弧中的内容可根据需要取舍。

<…>尖括弧中的内容由用户提供。

注意：在机器上练习各种语句时，花括号、方括号、尖括号并不敲入机内，而敲入括号中的实际内容。

/ 斜线代表或者的意思。

↓ 代表回车键，即键盘上的↓或 RETURN 或 ENTER 键。

∧代表键盘上的 CT L′键。

∧W 表示先按住 CTL 键不松手，再敲 W 键。

ESC 代表键盘上的 ESC 键。

在实习 dBASEⅡ之前，请参见键盘上的符号键。

1.3 常量、变量、函数及数据类型

1.3.1 常量、变量和函数

常量就是其值固定不能变化的量。也称常数。常量包括

数值常量、字符常量和逻辑常量。数值常量与数学中的常数相同。字符串常量是用引号将字符序列括起来，如："ABC"，"我是学生"。逻辑常量是逻辑值.T.或.F.（表示真或假）。

变量就是其值是可以变化的量。变量包括两种：一种是与数据库文件结构有关的字段名，这称为字段变量。如前面所述的学号、姓名、性别，都是字段名。另一种变量类似BASIC 语言中的变量，它用于存放操作中一些临时性的结果，称为内存变量。

变量名的构成方法同文件名一样，区别仅在于变量名最多允许 10 个字符。

dBASEⅡ提供了如下一些函数，其含义类似于 BASIC 语言中函数的概念。

函数　　功能

INT（<数值型表达式>）　取出值的整数部分

TYPE（<表达式>）　测定表达式的类型

VAL（<字符型表达式>）　将仅由数字组成的字符转换成数值

STR（<算术表达式，<长度>[，<小数位数>]）

LEN（<字符型表达式>）　将数值型转换成字符型

$（<字符型表达式>，<开始位置>，<长度>）

取子字符串。

@（〈字符表达式-1〉，〈字符表达式-2〉）

求字符表达式-1 的值在字符表达式-2 中的开始位置

!（<字符型表达式>）　将小写字母转换成大写字母

TRIM（<字符串>）　去掉字符串未尾的空白(格)

DATE ( )　　求系统日期
#　　求当前记录号
*　　测定记录是否注有删除标记
EOF　　判断文件是否处于结束
&　　替换出字符型变量的值

1.3.2 数据类型

数据类型是指常量、变量或函数的值所具有的类型。dBASEⅡ提供了三种数据类型：数值型、字符型和逻辑型。凡是取值为算术值的变量或常数，其数据类型均为数值型。凡是取值为字符的变量或常数，其数据类型均定为字符型。凡是取值为"真"或"假"的变量或常数，其数据类型都指定为逻辑型。

常量的数据类型是按常量的书写形式来确定的。如果按照数值常量的形式来书写，那么其类型是数值型的，其余类推。

字段的数据类型是根据字段名的实际取值情况，由用户在定义文件结构时用下面的字母来指定的：

C——代表字符型（Character)。如果某个字段名取值是字符（英文字符或汉字或空格)，那么必须用C来指明。如：姓名字段，其数据类型就一定是C。

N——代表数值型（Numeric)。如果字段值要参与算术运算，那么该字段就一定用N指明。如：年龄，奖学金等字段。

L——代表逻辑型（Logical)。如果字段取值.T.（真）或.F.（假)，那么就必须用L指明。

注意：有些字段既可指定为N，也可指定为C，例

如：学号字段就是如此。判定的准则是：若学号可能参与算术运算，则一定为 N 型。否则，尽管学号全部是数字，也仍然指定为 C 型。

在定义字段时，要涉及到字段宽度的这个概念。所谓字段宽度是指字段名所取值能够容纳的最大字符个数。例如："姓名"字段的宽度若为 8，那么就表示数据库中保存的所有人的名字不能超过 8 个英文字符（即 4 个汉字）。字段的类型和宽度一经确定，以后就只能按照这种类型和宽度来存取数据的值。

1.4 表达式

dBASEⅡ表达式可分为：算术表达式、关系表达式、逻辑表达式和字符串表达式。前三类表达式类同 BASIC 语言。

算术运算符有：+，–，*，/ 用于构成算术表达式。

加，减，乘，除

关系运算符有：>，>=，<，<=，<>或#。用于构成逻辑表达式。

大于，大于等于，小于，小于等于，不等

逻辑运算符有：.NOT.，.AND.，.OR.。用于构成逻辑表达式。

非，与，或

字符串运算符：+，–，$。用于构成字符串表达式。

字符串表达式是指：字符串常数或字符串变量或字符串函数，或用字符串运算符把上述各项连结而成的式子，都可以叫做字符串表达式。

+——把两个字符串直接连串起来，生成一个新的字符

串。

−——作用同+，区别仅在于：如果“−”号左边的字符串末尾有空白，则把空白移到结果的末尾。如：“abc___”+“def”→“abc___def”，而“abc___”−“def”→“abcdef___”其中_表示一个空白。

$——测定字符串中是否含有子串。规则是：若$的左端包含在右端之中，则运算的结果为真，否则为假。如“ABC”$“XYABCW”的结果为.T.（真）。

注意：

①可以用圆括号“(”和“)”来改变运算的优先顺序。

②表达式的值所具有的数据类型决定了表达式的类型。

③如果表达式中包含了多种运算，其运算的优先顺序是：算术运算，关系运算、字符串运算和逻辑运算。

习题一

1.举例说明数据库的特点。

2.在图 1.3（b）（c）中，哪些结点有两个双亲？各自的双亲是哪些结点？

3.举一个表格的实例，并说明其文件名、字段名以及它的记录。

4.结合 3 题，说明字段类型和字段宽度的含义。

5.如何给 dBASEⅡ的文件和字段取名？为什么文件名和字段名的字符间不能夹杂空格（白）？

6.常量、变量以及表达式的数据类型是如何确定的？分别举例说明之。

7.求下列字符串运算后的结果。

①“XYZ”$“XYZWZYZ”

② “XYZWXYZ” $ “XYZ”

③ “ABCD___”+“EFG”

④ “ABCD___”−“EFG”

第二章　cdBSEⅡ的安装和运行

本章介绍如何在中华学习机 CEC—Ⅰ上安装汉字 dBASEⅡ以及一些简单命令的运行。

2.1 系统要求

在中华学习机上运行 cdBASEⅡ，它需要一定的硬件和软件支持。

2.1.1 硬件环境

在中华学习机上运行 cdBASEⅡ，必须满足如下的硬件环境:

*1.*基于 *8080*、*8085* 或 Z_{80} 的微处理机系统。

在 CEC—Ⅰ上，必须插上一个 Z_{80} 卡。

*2.*至少 *64K* 字节的内存。

对西文 dBASEⅡ来说，至少需要 48K 字节内存。

*3.*显示器（可以用家用电视机代替）。

*4.*至少一个软盘驱动器。

*5.*最好有打印机。

出厂的中华学习机只留有一个插件槽，用于插 Z_{80} 卡后就不能接打印机。可选配五槽口扩展接口板解决本问题。

2.1.2 软件环境

除了硬件以外，还必须有软件的支持。

1.中文 CP／M 操作系统。

通常用 APPLEⅡ CP／M Ver 2.20B 代替。

2.cdBASEⅡ数据库管理系统。

目前，中华学习机 CEC—Ⅰ上使用的 CP／M 操作系统和 cdBASEⅡ都拷贝在同一张软盘上，由于 CEC—Ⅰ容量有限，只具有 CP／M 最核心的部分。这样作的目的是为了给用户留下较多的磁盘容量。

2.1.3 系统文件及其作用

我们把拷贝有 CP／M 操作系统和 cdBASEⅡ文件的软盘称为系统盘，其文件称为系统文件。要正确运行 cdBASEⅡ，系统盘上必须有如下一些系统文件:

文件名	扩展名	占用空间	作用
DBS	.COM	25K	主程序
DBASEMAⅠ	.OVR	7K	主覆盖程序
DBASEMSG	.COM	8K	处理出错提出信息
DBASEAPP	.OVR	4K	处理添加记录
DBASEBRO	.OVR	2K	处理浏览记录
DBASEJOI	.OVR	1K	处理联结记录
DBASEMOD	.OVR	3K	处理文件结构的修改
DBASERPG	.OVR	4K	处理报表文件
DBASESRT	.OVR	1K	处理合并细项

DBASEUPD .OVR 1K 处理更新文件

DBASEMSC .OVR 4K 其它

PIP .COM 操作系统程序

在整个系统文件中，PIP.COM 是操作系统的一个命令，其作用是负责装入外围交换程序，供以后的磁盘文件和外围传送操作之用。其余 12 个程序均是与 dBASE 有关的程序模块。dBASEⅡ系统成功的采用了覆盖技术，使得在如象中华学习机这种内存容量不太大的机器上仍然可以使用 dBASEⅡ数据库。从上表已经看到，12 个模块中就有 10 个模块是用覆盖技术来实现的（程序扩展名为.OVR 的是覆盖模块——DBASEMAI.OVR，它由 DBS.COM 调入，作为常驻模块，负责对输入的 dBASE 命令进行解释。如果输入进来的命令 DBASEMAI.OVR 能够处理，那么就立即进行处理。否则，立刻调用其它功能模块覆盖主模块，并即刻处理之，处理结束后再恢复被覆盖的主模块 DBASEMAI.OVR。如果输入或执行命令过程中发生了错误，则由 DBASEMSG.COM 调出相应的出错信息进行处理。其余模块的作用是显然的。在我们的使用中，已将上述文件复制到一张软盘上了。

2.1.4 Z_{80} 卡的插入方法

中华学习机 CEC—Ⅰ虽然具有 64K 字节的内存容量，但仍然不能直接在其上运行 cdBASEⅡ。因为 cdBASEⅡ要求主机芯片是 8080、8085 或 Z_{80} 的，而中华学习机 CEC—Ⅰ不满足这一要求（APPLEⅡ也如此）。为了不改动 CEC—Ⅰ的硬件，可以采用插上一个 Z_{80} 卡的方法来使之满足要求。如何插入 Z_{80} 卡？

1.小心打开中华学习机 CEC—Ⅰ的面板盖。(事先一定要拨下电源插头，断开电源)

2.将一个 Z_{80} 卡对准插件槽，然后平稳地插入到槽口内。(Z_{80} 卡上有指示灯的那一面对准用户。切记不要把方向弄反了)

3.由于 Z_{80} 卡占据面板盖位置，因而使用期间不能上面板盖。对于选配了五槽口扩展接口板的用户，不必取下面板盖。

2.2 cdBASEⅡ的启动

在使用 cdBASEⅡ之前，必须先将 cdBASEⅡ装入机内。根据主机电源事先是否已打开分为冷启动和热启动两种方式。

2.1 冷启动

所谓冷启动就是事先没有打开主机电源的情况下来启动系统。

1.请事先准备好系统软盘。其上有 cdBASEⅡ的 12 个程序和 CP／M 操作系统。

2.把系统盘插入到 A 驱动器中，如果只有一个驱动器，把系统盘直接插入就行了，关上驱动器的“门”。

3.打开屏幕电源后再打开主机电源。稍候，屏幕显示：

```
APPLEⅡ          CP/M
56K    Ver.     2. 20B
(C) 1980    MICROSOFT
A>
```

这是有关版权的提示，A>表示 CP／M 操作系统安装成功。

*4.*在 *A*>提示下，敲入：*DBS*↓。这时可以看见 Z_{80} 卡的指示灯在闪亮，同时听到读磁盘的声音。稍候，屏幕显示：

请输入日期（MM／DD／YY）或回车：

这时系统要求用户敲入日期或回车键。如果输入日期，必须按月／日／年的形式输入。也可以直接敲回车键↓。假设我们敲入回车键↓，屏幕显示：

请输入日期（MM／DD／YY）或回车：
汉字 DBASEⅡ　V1.0

出现一个小圆点，表示 cdBASEⅡ安装成功，这时就可以开始使用 cdBASEⅡ的各个命令了。如果有两个驱动器，可用 SET　DEFAULT　TO　B：↓设置默认工作盘。

2.2.2 热启动

所谓热启动就是在主机电源已经打开的情况下再来启动系统。热启动和冷启动类似，唯一的区别是将冷启动的第 3 步：打开主机电源改成：敲：PR#6↓，其效果和屏幕提示与冷启动相同。

注意：PR#6 是 CEC—Ⅰ机上热启动的操作标志。字

母必须用大写（压下 Caps Lock 键后显示的字母均是大写形式）。

2.3 汉字的输入方法

在 cdBASEⅡ的应用中，经常要用到汉字数据，因而先介绍汉字的输入方法。

汉字输入方式有两种：一是用汉语拼音；二是用区位码。不管是在 CP／M 状态下还是在 cdBASEⅡ状态下都可用上述两种方式中的任一种来输入汉字。

当我们希望在光标所处的位置上输入汉字时。先敲 F_2 或 L 键，使系统处于汉字输入状态。当敲 F_2 或 L 键以后，屏幕左下角出现“拼音区位”字样，表示系统正处于汉语拼音和区位码方式，这时既可以用汉语拼音方式输入汉字，也可以用区位码方式输入汉字。我们先介绍用拼音输入汉字。

2.3.1 汉语拼音输入法

例如：要输入“电视机”的“电”字。其操作过程是：

1.敲 F_2 或 L 键，使屏幕左下角出现“拼音区位”提示字样。

•

拼音–

区位–

2.敲入汉字的汉语拼音字母。

“电”字的拼音是“DIAN”，因而依次敲入字母：DIAN。当一个汉字的拼音多于4个字母时，只敲4个字母，不足4个字母时，敲一个空格键。屏幕显示如下：

·

拼音
DIAN 0 涤 1 翟 2 嫡 3 抵 4 底 5 地 6 蒂 7 第–
区位

3.敲空格键往下查找，敲键◁往回查找。

如果需要的汉字还没有出现在屏幕的下端，就敲空格键，直到所需汉字出现为止。如果跑过了头，敲键◁就会往回退。如果汉字已出现，则转第4步。

在上例中，尽管已敲入了DIAN，但“电”字仍然没有出现，待再敲2次空格键后，“电”字出现。屏幕显示如下：

·

拼音
DIAN 0 点 1 典 2 靛 3 垫 4 电 5 佃 6 甸 7 店–
区位

4.敲入数字号码键取字。

当所需汉字出现在屏幕下端后，敲入相应汉字前面的数字代码，就可以把该字取到先前的光标位置。“电”字的代码是 4，我们敲入数字“4”，“电”字就取到了。同时，显示出“电”字的区位码 2170。屏幕显示如下：

· 电–

拼音
　　2171　0 点 1 典 2 靛 3 垫 4 电 5 佃 6 甸 7 店–
区位

5.当输入汉字以外的字符时，必须退出汉字输入方式。

F_2 或 L 键进入和退出汉字输入的转换键。当系统已处于“拼音区位”方式时，再敲 F_2 或 L 就退出“拼音区位”方式，只有在这时才能输入非汉字字符。

6.继续输入另一个汉字。

可以在“拼音区位”下直接敲入拼音，也可以用 F_2 或 L 清除屏幕下端提示后再敲 F_2 或 L，使之处于新的“拼音区位”后再敲入拼音。

再强调一下，拼音多于 4 个字母时只敲入前 4 个字母，不足 4 个字母必须敲一个空格键。往下查找敲空格键，往回查找敲键◁。

2.3.2 区位码输入法

区位码输入法的关键是用户必须记住或知道汉字的区位码。当需要输入某个汉字时，只需在“拼音区位”提示下直接

敲入该汉字的区位码，系统就直接取到该汉字。因为一个区位码只对应一个汉字，所以不必象拼音那样用数字代码来选取。如：电视机的“电”字，当敲 F_2 或 L 后屏幕提示“拼音区位”时，再敲入 2171，“电”字就立刻取到先前的光标位置。

2.4 简单命令的用法

dBASEⅡ有两种工作方式。一是对话（一问一答）方式，其特点是只要在点状态下敲入一个命令，系统就立刻执行之，并得出结果。二是程序方式，其特点是：需要完成的操作必须以程序（当然可以直接在机器上敲入）的形式输入机内，待程序输入完毕，再用专门的命令执行之。两种工作方式各有千秋。对话方式的最大优点是直观、立竿见影。命令一旦输入完毕，机器立刻执行并将结果显示在屏幕上（如果有打印机，也可以打印在宽行纸上）。对初学者是方便的。最大的缺点是不能自动化，离不开人的干予，速度较慢。程序方式恰好克服了这些缺点。对实际应用和 dBASEⅡ的熟练者来说，程序方式更具魔力。下面先介绍一些简单的命令。

1.显示表达式的值

格式：? ／?? （表达式）

作用：计算表达式的值并在屏幕上显示出来。

例 1 ·? _5 * 6↓ （·是屏幕提示，不敲入。_表示一个空格）

30 （屏幕上显示 5 * 6 的值）

· （系统自动回到点状态）

例 2　·?? __5 * 6↓　　30　　(30 是屏幕上显示的 5 * 6 的值)

?和?? 的区别仅在于: ? 把计算结果显示在下行上,?? 把结果显示在同行上且可能盖掉?? (参见例 2)。在敲入命令时,? 或?? 之后必须至少敲一个空格再敲表达式 5 * 6,否则要出错。

2.保存表达式的值

如果要把一个表达式的值保存起来，以备后继命令中使用，就可用存贮命令 STORE。

格式: STORE〔表达式〕TO〔内存变量名〕

作用: 将表达式的值传送到内存变量名中保存起来，以备后用。

这里的表达式可以是一个具体的常数，或一个有值的变量名，也可以是一个表达式。常数、变量名、函数都是表达式中的特殊情形。

例 3　将 5 * 6 的值传送到内存变量 X 中。

·STORE__5 * 6__TO__X↓

30　　(系统将 5 * 6 的值 30 自动显示在屏幕上)

例 4　求 b^2-4ac 的值并送到 D 中。设: a=3, b=4, c=5

·STORE__3__TO__A↓

·STORE__4__TO__B↓

·STORE__5__TO__C↓

·STORE__B * B-4 * A * C__TO__D↓

·? __D↓

-44

·

例 5 将字符串保存在一个内存变量 W 中

·STORE_"学习和使用数据库技术。"_TO_W↓

·? _W↓

学习和使用数据库技术。

·

为了简洁起见，在以后的书写中，凡是命令中要敲空格的地方，我们都留出一个空白而不再用_标出，例如：STORE 5*6 TO X↓。读者在上机实习时，注意留出空白的地方必须至少敲一次空格键。

通过 STORE 命令就可以把表达式的值保存到内存变量中。一个内存变量一旦得到了值，就有了定义，也叫做定义内存变量。除了用 STORE 命令可以定义内存变量名以外，还有其它方法，以后将会看到。内存变量只能在内存中起作用，一旦断电，内存变量及其值也就不复存在了。在 dBASDEⅡ中，不同名的内存变量至多 64 个。

3.释放内存变量

如果程序或对话中用到不同的内存变量名超过了 64 个，就会妨碍操作的继续进行。解决内存变量个数超界的办法有二：一是重复使用变量名，即先前已定义的内存变量名，再重新定义赋与新的值。二是去掉一些内存变量名，这就是所谓的释放内存变量。

格式：RELEASE [<内存变量表>/ALL]

作用：释放指定的变量名。

其中：内存变量表是由一个或多个变量名构成。如果是多个变量名，那么各变量名彼此间要用逗号隔开。ALL 表示所有的意思，命令中若选用了 ALL 就表示释放所有的内

存变量名.

例 6　释放内存变量 X，A，B 的值。

· RELEASE　X，A，B↓

· ? X，A，B↓　　　(由于 X，A，B 已释放掉，就不能再显示其值)

(提示信息)

变量没找到。

例 7.释放所有内存变量。

· RELEASE　ALL↓

注意：方括号中的项称为待选项，根据用户需要而确定取舍。在释放命令中，如果不选用待选项，也表示释放内存变量名。内存变量一经释放，其名其值都不存在了。

4.存贮文件的建立和调用

前已述及，内存变量只能在内存中存在，一旦断电，所有值均会全部去掉。如果我们希望长期保存内存变量的值，就必须把内存变量名及其值组织成文件。在计算机中，只有用文件形式组织起来的内容才能脱离内存而永不消失。内存变量文件（或称为存贮器文件）是我们所要介绍的 7 种文件中最简单的一种，其建立的命令格式如下：

格式： SAVE　TO　〔文件名〕

作用： 将已有定义的内存变量组织成一个文件。

例 8.将内存变量 R、X 和 Y 组织到存贮文件中。

· STORE　5　TO　R↓

5

· STORE　5*8　TO　X↓

40

· STORE "学习数据库" TO　Y↓

学习数据库

· SAVE　TO　SFILE↓　　(SFILE 是存贮文件的名字)

通过 SAVE 命令，就把 X 和 Y 的值连同它们的名字一起保存到磁盘上的 SFILE 文件中去了。以后需要再用到 X 和 Y 的值，就可从 SFILE 文件中调用。调用存贮文件的命令是：

格式： RESTORE　FROM　〔文件名〕

作用： 将指定存贮文件中的内存变量名及其值调入内存。

例 9. 将 SFILE 文件中的内存变量调入内存。

· RESTORE　FROM　SFILE↓　　(调用存贮文件)

如何知道存贮文件中内存变量已调入到内存呢？可用显示内存变量的命令显示之。

格式： DISPLAY／LIST　MEMORY

作用： 在屏幕上显示内存变量的名和值。

例 10. 把从 SFILE 文件中调入的内存变量显示在屏幕上。

· DISPLAY　MEMORY↓　　(以下是屏幕显示)

R　　(N)　　5　　(变量名，类型，值)

X　　(N)　　40

Y　　(C)　　学习数据库

合计　　03　　变量被使用　　(用了 3 个变量)

00022　字节被使用　　(占用 22 个字节)

5.清屏幕

经过一系列的对活操作后，屏幕上的字符可能很零乱，

我们很希望象黑板上的字一样全部擦掉，这可以用清屏命令。

格式：ERASE

作用：清除屏幕上的所有字符，并将光标定位在屏幕的左上角（0行0列位置）。

例11.清除屏幕上的所有字符。

·ERASE↓　　（字符清除了，光标定位在左上角）

6.退出dBASEⅡ，回到操作系统状态

如果我们这次上机结束，或者希望回到操作系统状态之下，就要用到退出命令。

格式：QUIT

作用：无条件回到操作系统提示状态。

例12.退出数据库，回到A>状态。

·QUIT↓　　（以下是屏幕提示）

***END　RUN　CDBASEⅡ　***

A>　　　（结束运行CDBASEⅡ，回到操作系统）

注意：当退出cdBASEⅡ时必须用QUIT命令。这时系统自动关闭所有打开的文件，并将必要信息登录到磁盘上。如果退出dBASEⅡ回到A>后又想立刻进入dBASEⅡ，那么只需敲入：DBS↓就行了。

2.5. 技术规格和操作总则

cdBASEⅡ用起来简单和方便，但仍然有一些限制，实际应用中必须遵守。

2.5.1 技术规格

1.每个数据库文件至多65535个记录。由于出厂的CEC—Ⅰ只有一个驱动器，系统盘上所剩空间不多，实际上容纳不了这么多个记录。(可选用双驱动器接口板，扩充为两个驱动器)。

2.每个记录至多32个字段。

3.每个记录至多1000个字符（或500个汉字）

4.每个字符型变量至多容纳254个字符（或127个汉字）。

5.数值型数据至多10位有效数字。

6.数值的表示范围是$\pm 1.0\times 10^{-63}\sim \pm 1.8\times 10^{+63}$。

7.至多64个内存变量，占用总空间1536字节。

8.报表标题最长254个字符（或127个汉字）。

9.在求和（SUM）命令中，至多5个数值型字段。

10.可以同时打开16个文件，其中至多两个数据库文件。

11.一个数据库文件上至多允许建立7个索引文件。

2.5.2 操作总则

1.每个命令必须以命令动词作为开头。

2.每个命令的书写必须符合语法格式。

3.命令中的待选子句的顺序是任意的。

4.命令行的最大长度是254个字符（包含空格）。

5.若命令动词等关键字多于4个字符，可以只写前面的4个字符。但拼写必须正确。如DISPLAY可写成DISP。

6.一个命令一行敲不完，可以用分号结尾，以示下行为续行。

7.最好不用命令动词和关键字作文件名和变量名。

8.在机器上敲入命令行时，命令动词或关键字前后必须至少留一个空格且都必须用大写字母。

习题二

1.请面对一台中华学习机，练习 cdBASEⅡ的启动。

2.练习下列的输入。

学习和应用 CDBASEⅡ数据库技术。

3.说说？和 STORE 两个命令的异同点。

4.自己定义若干个内存变量，要求变量的数据类型分别包括数值、字符和逻辑型。再建立成存贮文件并调用之。

5.释放内存变量的命令 RELEASE 对存贮文件中的变量是否起作用？为什么？请用例子来试试看。

6.有人说：清屏幕命令 ERASE 执行后，不但把屏幕上的字符去掉了，同时也把内存变量清除了。这种说法对吗？为什么？

7.有人说：用 QUIT 命令来退出 dBASEⅡ，对磁盘上的文件来说是最安全的。这是真的吗？为什么？试试看。

第三章　数据库文件及其操作

数据库文件是 cdBASEⅡ最基本而又最重要的一种文件，很多文件和命令都和数据库文件直接有关。可以说，把数据库文件的建立、操作和使用方法搞清楚，其它文件也就不难弄懂了。

3.1　生成数据库文件的基本方法

对于不同的情况，可以用不同的方法来建立和生成数据库文件。数据库建立后，可以马上输入数据。

3.1.1 数据库文件的创立

1.创建数据库文件

我们要利用计算机来检索信息，那么首先必须存贮信息，而存贮信息的最好形式就是组织成数据库文件，用到的命令（也可以称为语句）格式如下:

格式: CREATE　　<文件名>

作用: 允许用对话的方式建立一个数据库文件，并将文件名字自动登录在磁盘目录中。

其中：CREATE 是建立数据库文件的命令标志，其中文含义是创立创建的意思。文件名是用户给数据库文件取的名字。下面用一个例子来说明 CREATE 命令的用法。

例 1.将如下的商品信息建立成一个数据库文件。

商品信息

货号	品名	单价	型号
120	自行车	179. 93	环球-2
145	电冰箱	1030. 96	北极星
138	电视机	1140. 00	美乐-9
127	收录机	715. 74	神笛牌
158	缝纫机	161. 26	家佳乐

在用计算机来建立有关商品信息的数据库之前，首先要确定以下几件事情:

①数据库文件的名字

文件的名字既要取得短小又要能够体现存贮信息的意义，使人一目了然。本例就用“商品”为数据库文件名。

②数据库文件的字段名

通常可以直接用表格的栏目名称作为字段名。但要注意，字段名太长超过 10 个英文字符或 5 个汉字时必须用缩写形式。本例就直接用栏目：货号、品名、单价、型号作字段名。

③字段名的类型

所谓字段名的类型是指字段对应的那一列取什么样的值：是数值吗还是字符？或者是逻辑。这必须根据实际情况和以后的应用来确定。在本例中，货号确定为字符型(C)，理由是以后对货号不可能进行加、减、乘、除运算；品名只能确定为字符型（C)；单价必须确定为数值型

(N)，且有 2 位小数；型号也只能是字符型（C)。本例中没有用到逻辑型（L)。

④字段宽度

前面第一章已经说过，字段宽度是指字段值中所容许的最大字符个数。这通常要根据实际应用来确定。本例中假定：货号最多 3 位；品名最长 6 个字符（3 个汉字)；单价最多 7 位数字，其中包括 1 个小数点位，而小数点之后有 2 位小数，实际只有 4 位整数；型号最多 6 位（3 个汉字)。系统规定：字符型字段的最大宽度为 254，数值型字段的最大宽度为 19，逻辑型字段的宽度固定为 1。

需要强调的是：如果我们是为应用单位设计数据库，那么必须征求用户意见，以用户要求为准。如果用户要求与 cdBASEⅡ规定的有矛盾，必须进行协调。

现在假设系统已处于点状态，可直接敲入：CREATE　商品↓　（以下是屏幕显示的情况)

· *CREATE*　　商品↓　　（以下是提示信息，要求用户回答)

输入记录结构如下：

字段　名字，类型，宽度，小数位数

001　—

光标在“—”处闪动，要求我们回答：第 1 个字段的名字、类型、宽度（长度)，如果有小数，还需要回答小数位数（即小数点之后有几位小数)。我们可以按照前面约定用敲键的方式进行回答。如敲入：货号，C，3 后系统又要求输入第 2 个字段，屏幕显示（汉字输入方法请参见第二章

2.3 节）如下：

```
字段      名字，类型，宽度，小数位数
001       货号，C，3
002        —
```

我们再按上述方式回答第 2 个字段，……。如敲入：品名，C，6↓……，直到全部字段定义完毕。为使读者看清敲键的全过程，我们再把定义过程显示如下：

```
·CREATE      商品↓
输入记录结构如下：
字段         名字，类型，宽度，小数位数
001          货号，C，3↓
002          品名，C，6↓
003          单价，N，7，2↓
004          型号，C，6↓
005          ↓                 （定义字段结束）
```

注意："字段名字，类型，长度，小数点位置"以及字段列下面的数字"001、002、003、004"等都是系统给出的提示信息，不要敲入，也不能由用户敲入。而"货号，C，3"等是必须由用户敲入的。目前，中华学习机 CEC—Ⅰ上使用的 cdBASEⅡ的提示信息还没有被全部汉化，为使不熟悉专业英文的读者看懂提示信息，我们已将相应的英文提示信息全部译成了中文，并提示在原英文的位置上。

2.装入数据

当光标处于一个新字段时，敲入回车键就表示字段定义完毕。本例中定义了4个字段，当光标在“005”后闪动时敲入↓，表示字段定义结束，屏幕就提示：

现在就输入数据吗？Y

询问是否立刻输入数据。如果敲入字母Y，就表示现在需要输入数据。如果敲入其它字符就表示不立刻输入数据，系统将文件结构保存起来后回到点状态，此时我们可以进行其它操作。本例中敲入Y，表示需要立刻输入数据，屏幕显示为：

```
RECORD#    00001    (记录号，系统自动提示)
货号           : _  :
品名           :    :
单价           :     :
型号           :    :
```

光标在第1个字段的第1个字符位置上闪动，这里我们可以依次输入记录值。冒号之间的位置表示了字段的宽度。现在依次敲入：

120 自行车 179.93 环球－2

第1个记录就算输入完成，再依次输入第2个记录，……，直到所需记录全部输完为止。

在输入数据时，如果敲入的字符占满了字段宽度所给的位置（冒号之间的位置），系统把光标自动跳到下一个字段，开始接收下一个字段的数据；如果冒号之间的位置没有被输入字符占满，那么必须敲一个↓键，强行使光标跳到下个字段。光标所处的位置就是当前需要处理的位置。

当光标处于一个新记录的第1个字段的第1个字符位置时，敲入↓或∧W，就可以结束数据的输入。这时系统自动将输入的数据及整个文件结构保存在磁盘上，然后回到点状态。这样，一个数据库文件就算建成了。本例中我们一共要输入5个记录，光标处在第6个记录时，敲入回车键↓，系统回到点状态。

如何才能知道建立的数据库文件是否正确呢？可以用显示命令显示之。

```
·USE      商品↓                 (打开文件)
·DISPLAY   ALL↓                 显示文件记录)
00001      120 自行车       179.93   环球-2
00002      145 电冰箱       1030.96 北极星
00003      138 电视机       1140.00 美乐-9
00004      127 收录机       715.74   神笛牌
00005      158 缝纫机       163.26   家佳乐
```

至此，一个完整的数据库文件就建成了。用同样的方法，可以建其它数据库文件。

3.误操作的纠正方法

在建库和装入数据的过程中，难免不敲错键码，我们把敲键和输入数据过程中所发生的错误统称为误操作。根据作者的教学经验，初学者　在以下几方面最容易出错:

①在文件名字中夹杂空格（白）

如将数据库文件名“商品”，误敲为“商　品”，用户的想法是在“商”和“品”之间留一个空格，看起来美观些，但好看却不中用。违反了文件的取名规则。改正的办法是重新敲入

正确的文件名，系统将既往不答。

②在字段名字中夹杂空格（白）

这也是系统规则所不允许的。发生这类错误，系统将给出更正错误的机会。例如：将“货号”误敲为“货　号”，屏幕提示：

字段	名字，类型，宽度，	小数位数
001	货　号，*C*，*6*↓	（货号间有一个空格）
错误字段		（提示字段有错）
001	—	（更正的机会）

我们重新敲入：货号，C，6↓就行了。凡是定义字段中发生的错误，如：应该敲逗号的地方误敲为其它符号，定义类型不用字母 C、N、L，宽度超界等等，系统都给出这种提示。

③小数位数给得不合适

系统要求含有小数的数值型字段的宽度至少比小数位数要多 2 位。例如：如果单价字段的小数是 2 位，那么字段宽度应大于或等于 4。下面的定义都是错误的：

003　　单价，N，4，3↓　　或 003　　单价，N，4，4↓

顺便再次强调：各类字段的最大宽度必须在系统规定的范围内。如果实际应用中某个字符型字段的宽度要超过 254 个字符，那该如何办？请读者考虑。

④在回答“现在就输入数据吗?”时误敲 Y 以外的键

我们的本意是想敲字母 Y 以便立刻输入数据，但由于初次上机，心情激动以至手指都有点不听使唤，无意间碰到

了 Y 以外的键，因而造成给系统以违心的回答，系统便回到点状态。挽回局面的办法是在点状态下打开文件后再敲入：APPEND↓，例如：

·USE　　　商品↓

·APPEND↓　　　　　(以下就可输入数据)

实际上，凡是希望对某一个数据库文件继续输入数据，均可用此形式。

⑤敲入的数据有误

当我们敲入后面的数据时才发现前面的字符有错，这是常有的事，不必惊慌。改正的办法是利用系统提供的控制键将光标移到出错处，重新输入正确字符就行了。

∧*E*——光标移到上一个字段。

∧*X*——光标移到下一个字段。

∧*S*——光标左移一个字符位置。

∧*D*——光标右移一个字符位置。

∧*V*——允许在光标处插入字符。

◁———光标向左回退一个字符。

∧*Q*——正在操作的文件不存盘，回到点状态。

∧*W*——文件存盘后回到点状态。

这些控制键，为我们上下左右移动光标提供了方便，同时也为修改数据失误提供了方便。但要注意：这些控制键只有在输入数据的过程中才有效，如果已经退出或回到点状态下，那么就“离柜不认”了，这种情况只好用专门的命令来修改，后面将会看到。

3.1.2 选取工作区和打开数据库文件

cdBASEⅡ的任何操作，都只有在一定的内存区域中才能进行。系统将内存分成主区和辅区两个部分，每一部分称为一个工作区。

1.选取工作区

如果在进行操作之前，我们希望将要进行的操作在某一个工作区内进行，那么就可以事先选定区域，这就是工作区的选取。

格式： SELECT <PRIMARY / SECOMDARY>

作用： 选择主区（PRIMARY）或辅区（SECOMDARY）作为操作命令的工作区。

例 2.选择主区为工作区。

· SELECT PRIMARY↓ （以后的操作就在主区内进行）

注意：

①如果我们没有明确选定工作区，那么就隐含说明使用的是主区。

②工作区一经选定，只要没有重新选择，其后的操作就一直在该区内进行。

③对于涉及到两个数据库文件的操作命令，如：JOIN、UPDATE（后面介绍）等必须选择工作区。

实际上，选取工作区就是为其后的 cdBASEⅡ操作划定一个活动场所。刚选定的工作区或正在使用的工作区称为活化工作区或当前工作区。

2.打开数据库文件

任何文件在使用之前都必须先打开，但各种文件的打开方式并不完全相同。打开数据库文件可用 USE 命令。

格式：USE [<文件名>]

作用：打开指定的数据库文件并把记录指针指向文件的第1个记录位置。

记录指针是cdBASEⅡ中极为重要的概念，它是系统用来标识记录位置的一个内部计数器。随着命令的执行和记录位置的推移，指针的计数值也在随之变化。因而，我们把指针正指向的那个记录称为当前记录或活化记录。

例3.在主区内打开"商品"文件。

```
·SELECT    PRIMARY↓      (选取主区)
·USE       商品↓          (打开文件)
·DISPLAY↓                 (显示第1个记录)
00001   120    自行车   179.93   环球-2
```

注意：

①一个数据库文件一经打开，就可对它进行操作，操作次数不限。

②在同一工作区内只能打开一个数据库文件，如果又打开了另一个数据库文件，先前的数据库就会自动关闭。

③若USE之后不带数据库文件名，则表示关闭当前打开的数据库文件。

④刚打开或正在使用（先前已打开）的文件称为活化文件。

3.1.3 复制数据库文件

如果先前一个文件也没有，那么要得到数据库文件就只好用CREATE命令来建立。如果事先建立了一些数据库文件，现在希望将某个数据库文件的某些行（记录）和列（字段）抽取出来，从而组成一个新数据文件，那么可以用复制命令。

*1.*复制数据库记录

格式：COPY TO<目标文件名> [<范围>] [FOR<条件>] [FIELDS<字段表>]

作用：将活化数据库文件复制到目标文件中。

其中：

<目标文件名>就是要新建立的数据库文件名。

<范围>用于指出数据库文件的哪些记录要复制到新文件中，它提供了三种可选用的方式：

<范围>
- ALL 表示所有记录
- NEXT n 从当前记录开始的下面n个记录。
- RECORD n 第n个记录。

在一个具体应用中，三种情况只能给出一种。

FOR<条件>对记录给出限制条件，表示满足条件的记录才被选取。条件可以是一个关系表达式或者逻辑表达式。

FIELDS<字段表>表示旧文件中的哪些字段要复制到新文件中。如果是多个字段的话，字段彼此间要用逗号隔开。

这里对<范围>、<条件>和<字段表>的解释，也适用于其它命令。看看下面的例子。

例 4.把价格低于 500 元的商品复制到 DSP 文件中。

·*USE* 商品↓

·*COPY TO DSP ALL FOR* 单价<*500*↓

00002 记录被拷贝 （提示信息，表示有两

个记录被复制了)

·USE　　DSP↓

·DISPLAY　ALL↓

00001　120　自行车　179.93　环球-2

00002　158　缝纫机　163.26　家佳乐

·

例 5.生成一个高价商品文件 GSP，文件中只需要品名及其单价。(假设价格大于 1000 元的为高价商品)

·USE　　商品↓

·*COPY　TO　GSP　FOR 单价 > 1000 FIELD 品名，单价*↓

·*00002　记录被拷贝*

·*USE　　GSP*↓

·*DISPLAY　ALL*↓

00001　电冰箱　1030.96

00002　电视机　1140.00

例 6.生成一个家用电器文件 JD。

·*USE　商品*↓

·*COPY　TO　JD　ALL　FOR "电" $ 品名*↓

00002 记录被拷贝

·*USE　JD*↓

·*DISPLAY　ALL*↓

00001　145　电冰箱　1030.96　北极星

00002　138　电视机　1140.00　美乐-9

·

注意以下几点:

①如果没有选用范围、条件，那么系统将全部记录复制

到目标文件中。当然可以只选用范围或者条件，也可以两者同时选用。

②如果不选用字段表，则所有字段复制到目标文件中，即是说，新文件和活化数据库文件有完全相同的字段。

③待选项彼此的先后位置顺序是无所谓的。

2.复制文件结构

如果我们并不需要数据库文件中的数据，而是欣赏文件的结构，那么可以用复制文件结构的命令。

格式： COPY STRUCTURE TO <目标文件名>

作用： 把活化数据库文件的结构复制到目标文件中去。

例 7.把商品文件的结构复制到 SSP 中。

· *USE* 商品↓

· *COPY STRUCTURE TO SSP*↓

当然，也可以用 APPEND 命令把数据添加到 SSP 文件中去。

3.生成结构文件

用 CREATE 命令建立的数据库文件，字段名是用户根据实际表格取定的，记录也是根据表格内容输入的，这是地地道道的数据库。cdBASEⅡ中还可以生成另一种数据库文件，其字段名是固定的且只有 4 个字段，这 4 个字段分别是：字段名、类型、宽度和小数位数，而每一个记录就是实际的字段名、类型、宽度和小数位数这些具体的值。这种数据库文件专门称为结构数据库文件，简称结构文件。产生结构文件的命令是：

格式： COPY TO <文件名>STRUCTURE EXTENDED

作用： 用活化数据库文件的结构来生成结构文件。

例 8.用商品文件的结构来生成一个结构文件。

· *USE　商品*↓

· *COPY TO CSP STRRCTURE EXTENDED*↓

00004 记录被拷贝

· *USE　CSP*↓　　　　(打开结构文件)

· *DISPLAY　ALL*↓　　　(以下是结构文件 CSP 的记录)

00001 货号	*C* ·	*3*	
00002 品名	*C*	*6*	
00003 单价	*N*	*7*	*2*
00004 型号	*C*	*6*	

从这个例子可以看到：商品文件的各个字段名、类型、宽度、小数位数成了 CSP 文件的记录，记录的个数刚好是商品文件的字段个数。由于 CSP 是一种数据库文件，因而，凡是适用于数据库文件的操作对结构文件也适用。我们可以用这种方法来修改数据库文件的结构，例如增加一些字段或减少一些字段或更改字段。

4.结构文件转换成数据库文件结构

要把结构文件转换成数据库文件结构，可用命令：

格式： CREATE <文件名> FROM <结构文件名>

作用： 将结构文件的记录转换为数据库文件结构。

例 9. 把结构文件 CSP 的记录转换成数据库文件 DSP。

· CREATE　DSP　FROM　CSP↓

这就把 DSP 建成一个数据库文件，只是它还没有记录而已。我们可用 APPEND 来添加记录。如：

· USE　DSP↓

· APPEND↓

RECORD#　00001

贷号　　　　　：　　　：
品名　　　　：　　　　：
单价　　　　：　　　　：
型号　　　　：　　　　：

这时我们就可以按照前面介绍的方法输入数据。

从前面介绍的命令可以看出，建立和生成数据库文件有多种方式，以后还有别的方式，究竟什么命令好，要根据具体情况来确定。总的原则是：凡是能够根据已有文件生成的数据库文件，就不必再用 CREATE 命令去从头建立。

3.2 查看数据库文件

数据库建好以后，可以用显示命令或列表命令查看其内容和结构。

3.2.1 显示数据库文件

前面示例中我们已经用到了 DISPLAY，这就是显示命令。

*1.*显示数据库记录

格式：DISPLAY[<范围>] [FOR<条件>] [OFF]

作用：在屏幕上显示活化数据库文件的记录。

其中范围、条件和字段表的含义同前面讨论的相同。若选用 OFF，就表示不显示记录号码。

例 10.找出价格高于 500 元的商品。

· *USE*　　　商品↓

· *DISPLAY*　　*ALL*　　*FOR*　　单价>*500*↓

00002　　　*145*　　　电冰箱　　　*1030.96*　　　北极星

```
00003     138     电视机     1140.00     美乐-9
00004     127     收录机     715.74      神笛牌
·
```

每个记录前面的序号（如：00002，00003，00004）就是记录号，这是输入记录时由系统按输入记录的先后顺序而编定的，这个记录号也可看成是记录在磁盘上的地址。只要我们知道了记录号，也就能够很快找到整个记录。

例 11.找出价格高于 500 元且带“电”的商品，不显示记录号码。

```
·USE    商品↓
·DISPLAY ALL FOR 单价>500. AND ."电"$品名 OFF↓
145    电冰箱    1030.96    北极星
138    电视机    1140.00    美乐-9
·
```

注意：

①如果命令中不选用范围或者条件，那么只显示当前记录。

例 12.不带任何待选项的显示。

```
·USE    商品↓            (当前记录是第 1 个记录)
·DISPLAY↓
00001   120     自行车     179.93     环球-2
·
```

②如果使用了 OFF，则不显示记录号码（见例 11）。

③屏幕上每次至多显示 20 个记录（西文 dBASEⅡ），中文 cdBASEⅡ 至多显示 9 个记录后暂停，等待敲入任何键后又继续显示后继记录。如果记录个数太少就看不出这个特

征。

④如果需要将显示的内容同时打印在宽行纸上，可以在显示命令之后跟上 TO　PRINT。

例 13　· USE　　商品↓

· DISPLAY　　ALL　　TO　　PRINT↓

如果用了 TO　PRINT，事先一定要连通打印机，否则会锁住系统，以至敲任何键均不起作用。如果出现这种情况，只好关机后重新启动。

2.显示数据库文件结构

如果只显示数据库文件结构，那么可以用显示文件结构的命令。

格式：DISPLAY　　STRUCTURE

作用：显示活化数据库文件的结构。

例 14.显示商品文件的结构。

· *USE*　　商品↓

· *DISPLAY*　　*STRUCTURE*↓（以下是显示内容）

文件的结构：商品.*DBF*

记录个数：*00005*

最后更改的日期：*06 / 23 / 89*

主区使用数据库

字段	名字	类型	宽度	小数
001	货号	*C*	*003*	
002	品名	*C*	*006*	
003	单价	*N*	*007*	*002*
004	型号	*C*	*006*	
＊＊合计＊＊			*00023*	

注意：这里，总的宽度比定义文件时多一个字节。本来应该 22（3+6+7+6＝22）个字节，现在显示的却是 23，多出的那个位置系统留作删除标记用（后面将要介绍删除标记*）。

3.2.2 列表数据库文件

cdBASEⅡ除了用 DISPLAY 显示数据库记录或结构外，还可以用 LIST 命令。

格式 1：LIST [<范围>] [FOR<条件>] [FIELDS<字段表>] [OFF]

作用：用卷动方式列出活化数据库文件的记录。

格式 2：LIST　　STRUCTURE

作用：用卷动方式列出活化数据库文件的结构。

其实，LIST 同 DISPLAY 的作用和用法是类似的，格式中除了命令动词 LIST 和 DISPLAY 以外，其它部份完全相同，它们在用法上的区别仅在于：

1. LIST 用卷动的方式不停地把所有记录显示完毕。

停留在屏幕上的只是最后的几个记录。

2. LIST 没有周期性的暂停。

当然，可以用敲∧S 键的办法使之暂停，再敲任何键又继续列表。

3.如果既不用范围也不用条件，那么 LIST 显示的是全部记录。

例 15.找出价格高于 500 元且带“电”的商品。

· *USE　　商品*↓

· *LIST ALL FOR 单价>500. AND.“电”$品名*↓

00002	*145*	电冰箱	*1030.96*	北极星
00003	*138*	电视机	*1140.00*	美乐-9

·

例 16.列出全部商品。

·*USE*　　商品↓

·*LIST*↓

00001	*120*	自行车	*179.93*	环球-2
00002	*145*	电冰箱	*1030.96*	北极星
00003	*138*	电视机	*1140.00*	美乐-9
00004	*127*	收录机	*715.74*	礼笛牌
00005	*158*	缝纫机	*163.26*	家佳乐

·

请读者自己给出 LIST　STRUCTURE 的例子。

3.2.3 浏览数据库文件记录

DISPLAY 和 LIST 命令最大缺点是显示或列表记录时人工无法干予，因而无法进行修改。但系统提供的浏览命令弥补了这种不足。

格式： BROWSE

作用： 浏览活化数据库文件的记录并可以进行修改。

例 17.浏览商品文件的记录并将电视机的货号改为333。

·*USE*　　商品↓

·*BROWSE*↓　　　　（以下是屏幕显示）

RECORD#：*00005*（当前记录号）

货号	品名	单价	型号
158	缝纫机	*163.26*	家佳乐

我们要修改电视机的货号，就敲∧R 控制键使光标跳到电视机所在的第 3 号记录，再用∧E（左移一个字段）或∧X（右移一个字段）把光标移到 138 处，将 138 改为 333 就行了。这时屏幕显示:

RECORD#: 00003

货号	品名	单价	型号
333	电视机	*1140.00*	美乐−9

利用下面的光标控制键，我们可以随心所欲的浏览或修改记录。

∧*E*——光标左移一个字段，若已到最左端，则移到上一个记录。

∧*X*——光标右移一个字段，若已到最后字段，则移到下一个记录的开头。

↓——光标右移一个字段，功能同∧*X*

∧*R*——光标跳到上一个记录。

∧*C*——光标跳到下一个记录。

∧*S*——光标左移一个字符位置。

◁ ——光标回退一个字符。功能同∧*S*。

∧*D*——光标右移一个字符位置。

∧*W*——将当前数据库文件存盘后回到点状态。

∧*Q*——当前数据库文件不存盘，即撤消修改回到点状态。

灵活运用这些控制键，可以在屏幕上迅速地把光标上下左右移动，以便浏览和修改。

3.3 定位记录指针

BROWSE 命令的优越性在于我们能够比较方便地移动光标，其实际效果相当于拨动了记录指针。但 BROWSE 中拨动记录指针是由敲∧R（向上拨动）和∧C（向下拨动）控制键来完成的，能否由专门的命令来控制？当然可以，这就是定位记录指针。

3.3.1 转到指定记录位置

利用转向命令，可以将指针定位到希望的记录上。

格式：GOTO　（<记录>／TOP／BOTTOM）

作用：将指针定位在活化数据库文件的指定记录位置上。

其中：

记录号——可以是数据库文件中的一个具体的记录号码，也可以是一个具有算术值的表达式，表达式的值就代表记录号。如：GOTO　3 表示指针定位到第 3 号记录上，使之成为当前记录。

TOP——表示数据库文件的头，即第 1 个记录。

BOTTOM——表示数据库文件的尾，即最后那个记录。

例 18.把指针定位到商品文件的末尾。

·USE　　商品↓

·GOTO　　BOTTOM↓

·DISPLAY↓

00005　　158　　缝纫机　　163.26　　家佳乐

·

例 19.在上题基础上，把指针定在开头。

·*GO　　TOP*↓　　　　(GOTO 可简写为 GO)

·*DISPLAY*↓

00001　　*120*　　自行车　　*179.93*　　环球–2

·

例 20.将第 3、4 两个记录显示出来。

·*USE*　商品↓

·*GO*　*3*↓　　　　(GO 3 也可省写成 3)

·*DISPLAY*　　*NEXT*　*2*↓

00003　　*333*　　电视机　　*1140.00*　　美乐–9

00004　　*127*　　收录机　　*715.74*　　神笛牌

·

3.3.2 跳跃定位指针

GO 的优点是可以转向任何记录。除此之外我们还希望能够以某个记录为准而前后移动指针。跳跃命令即可达此目的。

格式：SKIP [[+／–]<算术表达式>]

作用：在活化数据库文件中，指针按当前记录的位置前进（记录号码增大的方向）或回退若干个记录。

其中：

+——表示前进。这里的前进是指记录指针朝记录号码增大的方向拨动。

–——表示回退，即向记录号码减小的方向拨动指针。

<算术表达式>——表示前进或回退的记录个数。它一定是能够计算出数值的一个算术表达式。

+–可以省略。如果不带正负号，那么也表示前进方

向。

例 21 ·*USE* 商品↓

·*SKIP* +2↓ （前进 2 个记录位置）

记录：*00003* （当前记录是第 3 号记录）

·*DISPLAY*↓ （显示当前记录）

00003 *333* 电视机 *1140.00* 美乐−9

·*SKIP* −2↓ （回退 2 个记录）

记录：*00001* （回退到第 1 个记录）

·*DISPLAY*↓

00001 *120* 自行车 *179.93* 环球−2

·*SKIP*↓ （前进到下一个记录）

记录：*00002*

·*DISPLAY*↓

00002 *145* 电冰箱 *1030.96* 北极星

·

注意：若不带任何待选项，即只有 SKIP，则表示前进到下个记录；+或−仅表示前进或回退的方向。

3.3.3 带条件的定位命令

GOTO 和 SKIP 都是一种无条件定位。但实际应用中往往希望按照某种条件来拨动指针。这就要用到条件定位。

*1.*条件定位

格式： LOCATE [<范围>] FOR <条件>

作用： 在活化数据库文件中，把指针定位到满足给定条件的第 1 个记录位置上。

例 22.将指针指向价格高于 500 元的第一种商品。

·*USE* 商品↓

· *LOCATE　ALL　FOR　单价>500*↓

记录：*00002*

· *DISPLAY*↓

00002　145　电冰箱　1030.96　北极星

·

注意：这个命令找到的又是满足条件的第一个（不一定是文件的第 1 个记录）记录。如果需要找第 2 个满足条件的记录，那么必须用下面介绍的继续查找命令。如果给定范围内都没有满足条件的记录，那么系统提示“超出范围”或“没有找到”。

2.继续条件定位

格式：CONTINUE

作用：继续查找活化数据库文件中满足 LOCATE 命令所给条件的下一个记录。

例 23.找出价格高于 500 元的第二种商品。

· *USE　商品*↓

· *LOCATE　ALL　FOR 单价>500*↓

记录：*00002*

· *DISPLAY*↓

00002　145　电冰箱　1030.96　北极星

· *CONTINUE*↓　（继续查找）

记录：*00003*

· *DISPLAY*↓

00003　333　电视机　1140.00　美乐-9

·

注意：四个定位记录指针的命令（GO，SKIP，LOCATE 和 CONTINUE）都只能定出记录指针，不能显示

出记录值。如果需要把记录值显示出来，还得用 DISPLAY 命令（读者想想：用 LIST 是否恰当，为什么?）。

3.4 数据库文件的修改

读者可能还记得"离柜不认"的情景吧！如果在输入数据时没有把错误排除干净，那么只好用专门的修改命令了。对数据库文件的修改包括两个方面：一是修改数据；二是修改文件结构。

3.4.1 修改数据库记录

修改数据库记录的命令很多，如：EDIT，CHANGE，BROWSE，REPLACE。它们各有本领。

1.编辑命令

如果我们认定某个记录有错，那么用编辑命令也许是方便的。

格式：EDIT <记录号>

作用：允许用户用对活方式修改活化数据库中指定的记录数据。

例 24.将商品文件的第 5 个记录的价格改为 169.75 元。

· USE　　商品↓

· EDIT　　5↓　　　　（以下是屏幕显示）

RECORD#：00005

货号　　：158：

品名　　：缝纫机：

单价　　：　163.26：

型号　　：家佳乐：

修改过程是：用∧X 控制键或↓将光标下移到“单价”字段；再用∧D 将光标右移到“163.26”中的“3”处，将“3.26”改为“9.75”，敲∧W 将修改数据存盘后回到点状态。

在修改过程中，可以用如下的光标控制键：

∧*E*——光标上移一个字段，光标移到第一个字段时，则把当前记录存盘后再转到上一个记录继续编辑。

∧*X*——光标下移一个字段，光标移到最后字段时，则把当前记录存盘后再转到下一个记录继续编辑。↓也具有同样功能。

∧*S*——光标左移一个字符。◁具有同样功能。

∧*D*——光标右移一个字符。

∧*V*——允许在光标处插入字符。

∧*R*——将当前记录存盘后转到上一个记录继续编辑。

∧*C*——将当前记录存盘后转到下一个记录继续进行编辑。

∧*G*——删去光标所处的字符。

∧*Q*——撤消修改不存盘回到点状态。

2.改变命令

EDIT 不方便之处有二：一是不能指定字段；二是不能带条件。CHANGE 命令恰好克服了这两个毛病。

格式：CHANGE [< 范围 >] FIELDS < 字段表 > [FOR < 条件 >]

作用：允许用对话的方式修改活化数据库文件的记录。

例 25.修改商品文件的货号。

· *USE*　　商品↓

· *CHANGE*　*ALL*　*FIELDS* 货号↓

记录：*00001*

（第一个记录）

货号：*120*　　　　（原来的货号）
CHANGE? 120↓　　（把货号 120 修改成 115）
TO　　115↓　　（120 和 115 都由用户敲入）
货号：*115*　　　　（修改后的货号为 115）
CHANGE? ↓　　（询问是否还有修改，↓停止修改）

操作过程是：系统自动显示修改字段原来的值（120），CHANGE? …TO…的意思是把…修改成…。上例中“CHANGE? 120↓　TO　115↓”意思是把“120”改成“115”。CHANGE? 意思是询问用户还有修改吗？如果有，则重复上述过程，若无则敲↓结束修改。

现在再来显示商品文件的记录，可以清楚看见货号已被修改了。

· USE　　商品↓
· *LIST*↓

00001	*115*	自行车	*179.93*	环球-2
00002	*145*	电冰箱	*1030.96*	北极星
00003	*333*	电视机	*1140.00*	美乐-9
00004	*127*	收录机	*715.74*	神笛牌
00005	*158*	缝纫机	*169.75*	家佳乐

·

注意：如果命令中没有使用范围和条件，那么系统依次显示所有记录，等待用户修改。如果使用了范围或条件，那么系统自动显示满足条件的记录，等待修改后又继续显示满足条件的下一个记录，直到满足条件的记录被修改完或用↓强行结束修改为止。必须给出被修改的字段名。

3.浏览中修改

该命令的格式和用法请参见3.2.2。

4.替换命令

对于大批量且又有规律的修改，例如：所有商品降价5%，再用 EDIT 或 BROWSE 来修改显然就不方便了。对于这种情况，cdBASEⅡ提供了自动替换命令。

格式： REPLACE [〈范围〉]〈字段1〉WITH〈表达式1>[〈字段2〉WITH〈表达式2〉…][FOR〈条件>]

作用： 在活化数据库文件中，系统自动用给定的表达式的值来替换相应字段（WITH 左边的字段）的值。

例26.把所有商品降价5%。

·*USE* 商品↓

·*REPLACE ALL* 单价 *WITH* 单价*0.95↓

00005 条记录被替换　　　　（提示信息）

·*LIST*↓

00001	*115*	自行车	*170.93*	环球-2
00002	*145*	电冰箱	*979.41*	北极星
00003	*333*	电视机	*1080.00*	美乐-9
00004	*127*	收录机	*679.95*	神笛牌
00005	*158*	缝纫机	*161.26*	家佳乐

·

注意：如果不用范围和条件，那么就只替换当前记录；凡是能够用一个公式表示出来的情形，就可以用 REPLACE 来实现；表达式中可以包含字段名，也可完全没有字段名；表达式的类型必须与相应字段类型相同。

3.4.2 修改数据库文件结构

修改文件结构有三种情况：一是添加新字段；二是去掉字段；三是更改字段名。用到的命令是：

格式：MODIFY　STRUCTURE

作用：允许用对话的方式来修改活化数据库文件的结构。

这个命令有一个最大的缺点是修改文件结构时要破坏数据库文件中的记录，造成数据丢失。为了克服这一问题，可以这样来解决：

①首先将待修改文件复制一个新的结构备份，这个新结构是用于进行修改的。

②用 MODIFY　STRUCTURE 对新结构进行修改。

③将旧文件（待修改文件）的记录添加到新结构中生成新文件。再把新文件复制到旧文件。

④删去新文件。这时旧文件的结构就被修改好了且仍然保留了正确的数据。

1.添加新字段

如果数据库文件结构中需要新增加若干个字段，鉴于单独用 MODIFY　STRUCTURE 来修改文件结构要丢失数据，因而必须将下面命令联用。

格式：USE　<旧文件>　（打开待修改的文件）

　　COPY　TO　<新文件>　STRUCTURE

　　USE　<新文件>

　　MODIFY　STRUCTURE

　　{允许修改新文件结构}

　　APPEND　FROM　<旧文件>（把旧文件记录添加到新文件中）

COPY　TO　<旧文件>

DELETE　FILE　<新文件.DBF>（删去新文件）

我们用一个例子来说明这几个命令的联合用法。

例 27.在商品文件中添加一个"备注"字段。

·*USE*　商品↓

·*COPY*　*STRUCTURE*　*TO*　*NEWF*↓（复制结构）

·*USE*　*NEWF*↓

·*MODIFY*　*STRUCTURE*↓　（修改结构）

修改将删掉所有的数据记录＊＊＊＊＊＊继续？（*Y*／*N*）*Y*

	名字	类型	长度	小数
字段 *01*	：货号	*C*	*003*	*000*：
字段 *02*	：品名	*C*	*006*	*000*：
字段 *03*	：单价	*N*	*007*	*002*：
字段 *04*	：型号	*C*	*006*	*000*：
字段 *05*	：备注	*C*	*006*↓	：

修改过程是：敲入 MODIFY　STRUCTURE 后，屏幕提示"修改将删掉所有的数据记录＊＊＊＊＊＊继续？（Y／N）"意思是修改要破坏数据库中数据是否继续修改操作，回答 Y 就继续修改操作，回答 N 就不修改而回到点状态。由于我们先将数据库文件保护起来了，因而敲入字母 Y。这时屏幕显示出文件结构，光标处在"货号"位置。由于我们要添加字段，因而用↓或∧X 将光标下移到"字段 05"，再敲入"备注__C__006↓"，新添加字段定义完毕，敲∧W 结束定义，回到点状态。在敲入新字段名、类型和宽

度时，彼此之间要留至少一个空格。

紧接以上操作，将商品文件的数据添加到新文件中。

·*APPEND FROM* 商品↓

00005 记录被添加

·*COPY TO* 商品↓ （用新文件的结构和内容取代旧文件）

00005 记录被拷贝

·*USE*↓ （关闭 NEWF 文件）

·*DELETE FIEL NEWF.DBF*↓ （删去中间文件 NEWF）

商品文件是否修改了？我们可列出结构来看看。

·*USE* 商品↓

·*LIST STRUCTURE*↓

文件的结构：商品*.DBF*

记录个数：*00005*

最近更新日期：*06 / 23 / 89*

主区使用数据库

字段	名字	类型	宽度	小数
001	货号	*C*	*003*	
002	品名	*C*	*006*	
003	单价	*N*	*007*	*002*
004	型号	*C*	*006*	
005	备注	*C*	*006*	
* * 总计 * *			*00029*	

·

从这里可以看到，字段“备注”加上了。当然，我们也可以把旧文件复制到一个新文件中保存起来，再修改旧文件

(虽然失去了数据，但已经事先由新文件保存起来了，就不必担心)，等旧文件结构修改好了以后，再把新文件中保存的数据复制到旧文件中，其效果完全相同。请读者自己试试。

如果添加的字段不是在最后，而是在字段之间插入一个新字段，例如：要把“备注”字段插入到“单价”和“型号”之间，那么可以将光标移到“型号”字段，敲 N，屏幕空出一行，这时敲入新字段名就行了。

MODIFY STRUCTURE 命令中可用的光标控制键有：

∧*E*——光标上移一个字段。

∧*X*——光标下移一下字段。也可用↓键。

∧*S*——光标左移一个字符。也可用键◁。

∧*D*——光标右移一个字符。

∧*G*——删除光标所处字符。

∧*V*——允许在光标处插入字符。

∧*R*——满屏下移 8 个字段。

∧*C*——满屏上移 8 个字段。

∧*T*——删去光标所在行。

∧*N*——将光标所在行下移，留出一空行，允许用户插入新字段。

∧*W*——存盘后回到点状态。

∧*Q*——撤消修改回到点状态。

2.删除字段

方法同上，将光标移到待删字段，敲 T 键，字段被去掉，后继字段自动上移。请读者将添加的“备注”字段去掉。

3.更改字段名

在应用中，有时希望更改字段名中的某些字符。方法是：将光标移到待改字段，再把光标移到待改字符，重新敲入正确字符就行了。

需要特别强调的是：修改文件结构时一定要有效地保护好数据；在修改字段名时，不能改变字段的位置和宽度。因为一旦把位置和宽度变更，以后添加回去的数据和字段名就错位了。

对于字段的修改，除了使用 MODIFY STRUCTURE 命令以外，也可以用结构文件的方式来完成。请读者自己试试。

3.5 添加记录

添加记录有三种情形：一是在已知记录之间插入一个新记录；二是在文件末尾添加若干新记录；三是把一个文件接在另一个文件的末尾。下面分别介绍。

*1.*插入记录

格式： INSERT [BEFORE] [BLANK]

作用： 允许在活化数据库文件的当前位置插入一个新记录。

其中： BEFORE 是在…之前的意思，BLANK 表示空格（白）。

例 28. 在商品文件的第 2 个记录之后插入一个新记录：267 电饭锅 38.42 快熟牌

```
· USE      商品↓
· GO       2↓            (定位在第 2 个记录)
· INSERT↓                 (以下是屏幕提示)
```

```
RECORD#         00003
货号          :267:
品名          :电饭锅:
单价          :38.42:
型号          :快熱牌:
·LIST↓
00001    115    自行车    170.93    环球-2
00002    145    电冰箱    979.41    北极星
00003    267    电饭锅    38.42     快熱牌
00004    333    电视机    1083.00   美乐-9
00005    127    收录机    679.95    神笛牌
00006    158    缝纫机    161.26    家佳乐
·
```

插入记录的过程是：键入 INSERT 命令后，光标处在"货号"字段的第一个字符位置，我们按装入数据时所用的方法，敲入所需数据就行了。输入结束后敲↓键系统自动把新记录存盘。

注意：①INSERT 中可用的光标控制键有：

∧*E*——光标上移一个字段。

∧*X*——光标下移一个字段。

∧*S*——光标左移一个字符。也可用键◁。

∧*D*——光标右移一个字符。

∧*V*——允许在光标处插入字符。

∧*R*——把当前记录存盘后回到点状态。也可用∧*W*。

∧*Q*——不存盘回到点状态。

②若选用 BEFORE，则表示在当前记录位置上插入，原记录自动下推；否则在当前记录的下一个记录位置上插

入。

③若选用 BLANK，则由系统自动插入一个空白记录。

④若只有一个记录，INSERT 无效。

⑤INSERT 不能用于在文件末尾添加记录。

2.添加记录

格式： APPEND [BLANK]

作用： 允许用对话的方式在活化数据库文件的末尾添加若干新记录。

例 29. 在商品文件末尾添加新记录。

```
· USE     商品↓
· APPEND↓
RECORD#    00007
货号      ：149：
品名      ：电热褥：
单价      ：88.77：
型号      ：助温牌：
```

这同输入数据完全一样。用 APPEND 添加数据非常方便。当光标处在一个新记录的第一个字段的第一个字符位置时敲↓或 W 就结束添加数据而回到点状态。

APPEND BLANK 含义是自动添加一个空白记录，这在格式修改数据库文件时常用到。APPEND 命令中所用控制键同 INSERT 相同。

3.添加一个文件

格式： APPEND FROM <文件名>

作用： 将指定文件添加到活化数据库文件的末尾。

例 30. 假设采购了一些新商品，其信息保存在 NEWSP

文件中，现在希望将 NEWSP 文件的记录连接在商品文件的末尾。假设 NEWSP 文件中的记录如下所示：

货号	品名	单价	型号
139	电子琴	357.29	自学-4
183	钢琴	2300.00	新星-1

```
· USE      商品↓
· APPEND   FROM     NEWSP↓
00002 条记录被加入
· LIST↓
00001    115    自行车    170.93     环球-2
00002    145    电冰箱    979.41     北极星
00003    267    电饭锅    38.42      快熟牌
00004    333    电视机    1083.00    美乐-9
00005    127    收录机    679.95     神笛牌
00006    158    缝纫机    161.26     家佳乐
00007    149    电热褥    88.77      助温牌
00008    139    电子琴    357.29     自学-4
00009    183    钢琴      2300.00    新星-1
·
```

由上可以看到，NEWSP 文件的记录已连接到商品文件中去了。需要注意：NEWSP 文件和商品文件必须相同的结构（字段同名、同类型、同宽度）。如果结构不完全相同，那么只有同名同类字段值被连接。命令执行后，NEWSP 文

件的记录仍然存在。

3.6 删除数据库文件记录

数据库经过长时间使用以后，可能有些数据已不再需要了，这就可以删除之。

*1.注删除标志 **

对数据库中的数据，即使暂时不用，也不立刻去掉。因为一经去掉就不能再恢复，到这时就追悔莫及了。因而通常采用注删除标志的办法，使之有再“生”的机会。

格式：DELETE [<范围>][FOR<条件>]

作用：对活化数据库文件中指定的记录注上删除标志 * 。

例 31.给商品文件中带“琴”的商品注上删除标记。

```
·USE    商品↓
·DELETE   ALL   FOR   “琴”$  品名↓
00002 条删除
·LIST↓
00001    115    自行车    170.93    环球-2
00002    145    电冰箱    979.41    北极星
00003    267    电饭锅    38.42     快熟牌
00004    333    电视机    1083.00   美乐-9
00005    127    收录机    679.95    神笛牌
00006    158    缝纫机    161.26    家佳乐
00007    149    电热褥    88.77     助温牌
00008   *139    电子琴    357.29    自学-4
00009   *183    钢琴      2300.00   新星-1
```

·

货号 139 和 183 之前的“*”就是删除标志。

注意：如果不带范围和条件，则只对当前记录作删除标志。

2.去掉删除标志

如果我们希望恢复注有删除标志的记录，则可以用恢复命令。

格式： RECALL [<范围>] [FOR <条件>]

作用： 去掉活化数据库文件中记录注有的删除标志 *。

例 32.去掉商品文件中记录之前注有的删除标志 *。

```
· USE      商品↓
· RECALL     ALL↓
00002 条记录被恢复
· LIST↓
00001    115    自行车    170.93     环球-2
00002    145    电冰箱    979.41     北极星
00003    267    电饭锅    38.42      快熟牌
00004    333    电视机    1083.00    美乐-9
00005    127    收录机    679.95     神笛牌
00006    158    缝纫机    161.26     家佳乐
00007    149    电热褥    88.77      助温牌
00008    139    电子琴    357.29     自学-4
00009    183    钢琴      2300.00    新星-1
·
```

执行恢复命令后，删除标志不见了。注意：RECALL 命令的执行并不是去掉注有删除标志的记录，仅是把 * 去掉了。

3.压缩记录

如果真的希望把无用的记录去掉，那么可以用压缩记录的命令。

格式：PACK

作用：将活化数据库文件中注有删除标志的记录从库中清除掉。

例33.把带"琴"的商品去掉。

```
·USE    商品↓
·DELETE    ALL    FOR  "琴"$ 品名↓
00002 条删除
·PACK↓
压缩完成 00007 记录被拷贝
·LIST↓
00001    115    自行车    170.93     环球-2
00002    145    电冰箱    979.41     北极星
00003    267    电饭锅    38.42      快熟牌
00004    333    电视机    1083.00    美乐--9
00005    127    收录机    679.95     神笛牌
00006    158    缝纫机    161.26     家佳乐
00007    149    电热褥    88.77      助温牌
·
```

用 PACK 压缩掉的记录，在数据库文件中就再也找不到了。因此，使用 PACK 命令必须十分小心。

4.删除文件

如果整个文件都要去掉，那么可以用删除文件的命令。

格式：DELETE   FILE <文件名.扩展名>

作用：删去指定的文件。

例 34.删去 NEWLP 文件。

· DELETE　NEWSP.DBF↓

·

这样，NEWSP 文件就被删除了。需要注意的是：

删除的文件不能是打开的文件；文件必须带扩展名；用该命令可以删除磁盘上的任何文件。实际上，删除文件并不是一个一个地去掉记录，而是仅抹掉磁盘目录中的文件名就行了。

3.7 统计命令

在应用中，我们经常要统计记录的个数或将某一列的值全部累加起来，就可用统计命令。

1.计数

格式： COUNT [<范围>] [FOR<条件>] [TO<内存变量>]

作用： 统计活化数据库文件中记录的个数。

例 35.统计第 3～6 个记录之间价格高于 500 元的商品种数。

· USE　商品↓

· 3↓　　(定位当前记录)

· COUNT NEXT 4 FOR 单价>500 TO W↓

计数=*00002*

· ? W↓

2

·

注意： COUNT 命令是按行来计数的，统计的结果可

以保存在内存变量中，这也是定义内存变量的一种方法。命令中如果不带条件和范围，则表示对所有记录计数（相当于用 ALL）。

2.按列求和

如果我们希望把数据库文件中某字段的所有数值累加起来，可以用求和命令。

格式： SUM <数值型字段表>[<范围>][FOR<条件>][TO<内存变量表>]

作用： 对活化数据库文件中记录的字段按列求代数和。

例 36. 假设有如下的工资表格：

姓　名	基本工资	福利费	奖金	房租费	水电费	实发工资
张森林	73.46	2.53	7.00	1.89	2.13	
徐荣策	68.35	3.10	7.50	1.45	1.55	
吕逑才	68.92	3.00	8.00	2.11	1.98	
王锡刚	88.87	3.56	8.88	1.48	2.60	
陈　力	125.6	3.55	8.66	1.99	2.98	
总　计						
平　均						

要求用 cdBASEⅡ来求实发工资、各项总计值和平均值并填到相应的表格位置。

分析：先按表格给出的栏目建立工资数据库文件；输入数据，实发工资部分敲↓跳过，以后用 REPLACE 命令来替换。最后两个记录只输一个“总计”和“平均”，其它位置敲↓。（假设工资文件已按要求建好，注意字段宽度）

下面给出求实发工资及总计和平均的命令（内存变量 S_1、S_2、S_3、S_4、S_5、S_6 分别用于存放所有人的基本工资、福利费、奖金、房租费、水电费、实发工资的累加和。）

· USE　工资↓

· REPLACE　ALL　实发工资　WITH　基本工资+福利费+奖金--房租费—水电费↓

00005 条替换了

· GO　1↓

· SUM　NEXT　5　基本工资，福利费，奖金，TO S_1，S_2，S_3↓

· GO　1↓

· SUM NEXT 5 房租费 水电费 实发工资 TO S4, S5 S6↓

· GO 1↓

· COUNT　NEXT　5　TO　C↓

· GO　BOTTOM↓

· REPLACE 基本工资　WITH S_1/C，福利费 WITH S_2/C，奖金 WITH S_3/C↓

· REPLACE　房租费 WITH S_4/C，水电费 WITH S_5/C，实发工资　WITH S_6/C↓

00001 条替换了

· SKIP　−1↓

· REPLACE　基本工资　WITH　S_1，福利费 WITH S_2，奖金　WITH S_3↓

· REPLACE　房租费　WITH S_4，水电费 WITH S_5，实发工资　WITH S_6↓

00001 条替换了

· LIST　OFF↓

张森林	73.46	2.53	7.00	1.89	2.13	78.97
徐荣策	68.35	3.10	7.50	1.45	1.55	75.95
吕速才	68.92	3.00	8.00	2.11	1.98	76.63

王锡刚	88.87	3.56	8.88	1.48	2.60	97.23
陈　力	125.67	3.55	8.66	1.99	2.98	133.01
总　计	425.27	15.74	40.84	11.14	461.79	461.79
平　均	85.05	3.14	8.16	2.22	92.35	92.35

请读者认真理解这个实例中用到的各个命令。在 SUM 中为什么要用 NEXT 5? 如果换成 ALL，那么达到同样目的要作哪些修改? 请读者自己来完成。

需要强调的是：凡是按横行求和只能用 REPLACE 命令；凡是按列求和可以用 SUM 命令；所有的求和都只能针对数值型字段。

3.8 两个数据库文件的连接

如果我们希望把两个有联系的数据库文件的某些行和列拼成一个新的文件，那么可以用 cdBASEⅡ提供的连接命令。

格式：JOIN TO <文件名>FOR<条件>
[FIELD<字段表>]

作用：将主区（P.）和辅区（S.）内已经打开的数据库文件按条件进行连接，从而生成一个新的数据库文件。

例 37. 假设有学生和成绩两个数据库文件，其格式和数据如下：

学生文件

学号	姓名	性别	年龄	籍贯
89101	张正	男	16	成都
8908	李扬	女	15	北京
8903	兰兰	女	16	长春
8909	冬冬	男	14	成都

成绩文件

姓名	数学	物理	化学	语文	政治	总分
李杨	85	80	75	83	80	
冬冬	90	88	78	83	85	
张正	80	85	83	89	82	
兰兰	75	72	84	88	84	

假定两文件均建立且装入了数据。成绩文件的总分字段值输入时可敲↓跳过。现在要求找出成都籍学生各自的姓名、性别和总成绩。

· *USE* 成绩↓

· *REPLACE ALL* 总分 *WITH* 数学+物理+化学+语文+政治↓

00004 条记录替换了

· *SELECT SECONDARY*↓ （选择辅区）

· *USE* 成绩↓

· *SELECT PRIMARY*↓ （选择主区）

· USE 学生↓

· JOIN TO 学生成绩 FOR 姓名=S. 姓名 AND. 籍贯

="成都" FIELDS 姓名，性别，S. 总分↓

· USE　　学生成绩↓

· LIST↓

00001	张正	男	419
00002	冬冬	男	424

·

JOIN 命令的执行过程是这样的：

学生文件的第一个记录依次同成绩文件的每一个记录按给定的条件进行比较，如果条件成立，那么就把满足条件的两记录合起来生成一个新记录，作为新文件的第一个记录；如果条件不成立，则学生文件的第一个记录继续同成绩文件的下一记录比较，…，这样直到成绩文件的最后那个记录在比较过程中，条件成立则生成新记录，不成立就继续比较。然后，学生文件的第二个记录又同成绩文件的每一个记录进行比较…，如此重复以上过程，直到学生文件的最后记录。这就是说，两文件的记录每两两都要比较完。如学生文件有 4 个记录，成绩文件也有 4 个记录，那么 JOIN 命令要比 4 ×4＝16 次。由此可见，执行 JOIN 命令是很费时的。

JOIN 命令的执行情况如图 3.1 所示。

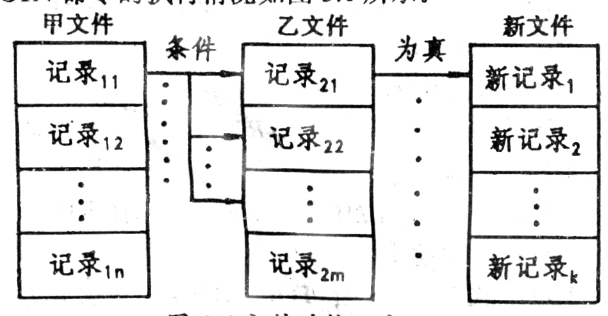


图 3.1 文件连接示意图

使用中需要注意：

①由于两个文件的每两个记录彼此都要进行比较，如果条件放得太松，产生的新文件记录个数有可能超过文件允许的限度。

②如果主区和辅区内使用相同的字段名，那么必须在同名之前缀以 *S*.或 *P*.，以示区别。通常在主区中引用辅区内的字段名，则字段名之前缀以 *S*.（如：*S*.姓名）。在辅区中引用主区内的字段名则前缀以 *P*.。

③如果命令中省略了 *FIELD* 待选项，那么新文件中的字段由主区和辅区两个文件的字段共同组成。选用原则是：首先使用主区文件的字段名，再使用辅区文件的字段名，直到最多 *32* 个字段为止。

习题三

1.建立如下三个数据库文件（自己给出字段类型、宽度和小数位数），并装入适当的数据。

学生（学号，姓名，性别，年龄，籍贯，奖学金）

课程（课号，课名，学时数，先修课）

学生选课（姓名，课名，成绩）

2.在上题装入数据的过程中，考察 3.1.1 给出的光标控制键。哪些键在你的机器上是有效的？

3.举例说明必须选择工作区的情况。

4.CREATE 和 COPY 命令都可用于建立数据库文件，它们在用法上有什么不同？

5.什么是结构文件？它的实际用处是什么？

6.COPY 命令的各种不同形式之间有什么区别？它们各适用于什么场合？

7.LIST 和 DISPLAY 命令的作用是相同的，都是用于显示数据库文件的记录或结构，因而 LIST 命令可以取代 DISPLAY，这个说法是否正确？为什么？举一个实例来说明。

8.BROWSE 也可用于显示记录，它的最大优点是什么？

9.GOTO、SKIP 和 LOCATE 都可用于定位记录指针，三者在用法上有什么不同？

10.EDIT、CHANGE 和 REPLACE 各适用于哪些情况？

11.在课程文件中增加一个字段：任课教师，请写出语句（命令）。

12.INSERET 和 APPEND 的区别是什么？APPEND 和 APPEND FROM <文件名>的区别又在哪里？

13.任何两个数据库文件都可用 APPEND FROM <文件名>命令连接在一起吗？试试看有什么问题？想想这是什么原因？

14.为什么 cdBASEⅡ中删除记录要采取先注标记 * 后用 PACK 来删去记录的办法，而不立刻直接删法记录？

15.在哪些情况下宜使用 DELETE FILE 命令？

16.在 cdBASEⅡ中，按行求和用什么命令？按列求和用什么命令？举一个实例来说明并写出语句。

17.在第一题建立的数据库文件中，找出选修了数据库的学生名字以及数据库课程的先修课．请用语句写出来。

18.如果对数据库文件希望按行来操作，那么相应命令中可以给出哪两个待选项？按列来操作呢？

第四章　函数与运行特征

cdBASEⅡ中配备了多种函数，按照函数值的类型可分为数值函数，字符处理函数以及逻辑函数。另外，本章还要介绍为方便用户而提供的运行特征控制。

4.1 函数及其用法

cdBASEⅡ的函数可以实现通常表达式难以实现或不能实现的运算。函数本身就是一种表达式，当然，函数也可以和常数、变量一起组成更为复杂的表达式，函数经过运算以后也一定能够得出一个值。

4.1.1　数值函数

所谓数值函数就是函数运算后得出的值是一个数值，这类函数称为数值函数。

1. **取整函数** *INT*

格式： INT(〈数值型表达式〉)

作用： 取出数值型表达式值的整数部份，把小数部份全去掉。

例 1. ? INT(158.723) ↓

158　　　　(只取整数部份)

·STORE　123.748　TO　X↓

·? INT(X) ↓

123

·? INT(-5 * 6／4) ↓

-7

注意：INT 函数不是四舍五入取整。

2.　字符串变换成数值 VAL

格式：VAL(〈字符型表达式〉)

作用：将字符型表达式值中左边连续数字符号转换成数值。

例 2. ·STORE　"1989.6"　TO　X↓

1989.6　　　　(字符　1989.6)

·?　VAL(X) ↓

1989.6　　　　(数值　1989.6)

·?　VAL("35AB43") ↓

35

·?　VAL("123456")

123456

注意：VAL 函数只能转换仅由数字组成的字符串。

3.求字符串长度函数 LEN

格式：LEN(〈字符型表达式〉)

作用：求出字符型表达式值中包含了多少个字符。

例 3. ·?　LEN("ABCDE") ↓

5

·STORE　"123456"　TO　X↓

·?　LEN(X) ↓

6

4. 求子串位置函数@

格式：@(〈子字符串〉,〈主字符串〉)

作用：求出子字符串在主字符串中的开始位置。

例4. ·? @("ABC","STWABCXY") ↓

4

·? @("数据库","*cdBASE*数据库技术") ↓

7

·*STORE* "我们的祖国象花园" *TO* *X*↓

·*STORE* "祖国" *TO* *Y*↓

·? @ (*Y*,*X*) ↓

7

5. 求当前记录号函数#

格式：#

作用：求当前记录的号码。

例5. ·USE 商品↓

·? #↓

1 (当前是1号记录)

·GOTO BOTTOM↓

·? #↓

7 (当前是第7号记录)

注意：数学函数运算后得到的值均是数值，可以直接参与算术运算。

4.1.2 字符处理函数

字符处理函数的最大特点就是经过函数运算后得出的值都是字符。

1. 将数值转换成字符串的函数 *STR*

格式：STR（〈数值型表达式〉，〈长度〉〔，〈小数位数〉〕）

作用：将数值型表达式的值按指定的长度转换成字符串。

其中：小数位数表示转换成字符串以后串中应保留的小数位。

例6. ·？ STR(123.456,7,3) ↓

123.456 （这是字符串123.456,相当于"123.456"）

·STORE 5*8 TO W↓

·？ STR(W*2.21,5,2) ↓

88.40 （字符串）

2. 将数值变成ASCII码字符CHR

格式：CHR（〈数值型表达式〉）

作用：将数值表达式的值转换成相应的ASCII码字符。

例7. ·？ CHR(78) ↓

N （字母N的ASCII码是78）

·？ CHR(10*8) ↓

P （字母P的ASCII码是80）

3. 取子串函数$

格式：$（〈字符型表达式〉，〈开始位置〉，〈个数〉）

作用：把字符型表达式值中从〈开始位置〉起的指定〈个数〉的字符取出来，生成一个子串。

例8. ·？ $（"ABCDEFGH",4,3） ↓

DEF

·STORE "我们的祖国象花园" TO W↓

·? $ (W,7,4) ↓

祖国

4. 日期函数 DATE

格式：DATE()

作用：产生一个载有日期信息的字符串。

当我们启动 cdBASE 时，系统要求我们按月／日／年形式输入日期，这个日期值就是保存在 DATE() 函数中的。

例 9. ·? DATE() ↓

06／23／89　　　　(表示 89 年 6 月 23 日)

5. 测试数据类型函数 TYPE

格式：TYPE(〈表达式〉)

作用：测试表达式值的类型．C—字符型，N—数值型，L—逻辑型。

例 10. ·? TYPE("AB"+"CD") ↓

C (表示"AB"+"CD"的值"ABCD"是字符型)

·STORE　5*8　TO　S↓

·? TYPE (S) ↓

N　　　　(数值型)

6. 删除末尾空格函数 TRIM

格式：TRIM(〈字符型表达式〉)

作用：去掉字符型表达式末尾的空格。

例 11. ·? LEN (TRIM("ABC　　　　")) ↓

3　　　　(ABC 之后的 4 个空格被去掉了)

7. 宏替换函数&

格式：&〈字符型内存变量名〉

作用：替换出内存变量的值。

例 12. ·*STORE*“商品” *TO* *B*↓

·*USE* *&B*↓ (相当于直接写“商品”)

·*DISPLAY*↓

00001 *115* 自行车 *170.93* 环球−2

·*STORE* “*” *TO* *CC*↓

·*?*“*75CC 80*”↓ (CC 和 80 之间留一个空格)

75*80 (将 CC 中的“*”替换出来)

·STORE “3333” TO C↓

·? 4444+&C↓

7777 (数值)

这个示例说明：&不但能够将字符替换出来放到所在位置上，而且能够将全由数字组成的字符串替换出来后转换成数值并参与算术运算。注意：&和其后的变量名之间不能留空格。如&C 是错误的。

4.1.3 逻辑函数

逻辑函数的运算结果一定是逻辑值真（用.T.表示）或假（用.F.表示）。

1. 检查文件是否结束函数 EOF

格式： EOF

作用： 若记录指针指向数据库文件最后那个记录之后的位置时，EOF 的值为真，否则为假。

例 13. USE 商品↓

·? EOF↓

.F. (当前记录是 1 号，所以 EOF 为假)

·GO BOTTOM↓

·SKIP↓ (下推一个记录位置)

·?　EOF↓

.T.　　(文件已经结束)

EOF 是 End of Filc (文件结束) 的缩写。这个函数在对文件的循环处理中很有用。

2. 判断文件是否存在函数 FILE

格式：FILE(〈字符型表达式〉)

作用：若字符型表达式的值代表的文件已在磁盘上，则函数的值为真，否侧为假。

例 14. ·?　TYPE("商品")↓

.T.　(表示商品文件已在磁盘上)

·?　TYPE("ABCD")↓

.F.　　(表示没有 ABCD 文件)

4.2　运行特征的设置方法

在执行 cdBASEⅡ命令的过程中，用户可以根据自已的需要，设置一些工作状态，如：是否响铃、是否暂停等等，这就叫做设置运行特征。如果用户没有明确地指出运行特征，那么系统就隐含一种特征，这称为标准状态。

系统用于设置运行特征的方式有两种：

格式：SET〈参数〉ON／OFF

作用：将某种状态设置为开 (ON) 或关 (OFF)。

格式 2：SET〈参数 1〉TO〈参数 2〉

作用：将参数 1 置为参数 2。

灵活地设置运行特征，将会给我们的应用带来极大的方便。

4.2.1 SET〈参数 ON / OFF〉的用法

SET〈参数〉ON / OFF 形式的运行特征共有 17 个，现分别介绍之。

1. SET BELL ON / OFF——是否响铃

数据填满字段的宽度时是否响铃。ON 表示要响铃，OFF 表示不响铃。如果在输入和修改数据时，敲入的数据字符填满给定字段宽度（即两个冒号: 之间的位置）时，希望系统用声音提醒我们，那么在操作之前就设置 SET BELL ON。系统隐含的标准特征是 ON。

2. SET CARRY ON / OFF——是否将前一记录复制到下一记录

在输入数据的过程中，往往有这种情况：后一个记录和前一个记录的内容大部分相同，这时不必重输后一个记录，而用 SET CARRY ON 特征将前一个记录内容自动复制到后一个记录中，再来修改不同部分。如用 APPEND 来添加数据，如果记录的某些部分是相同的，那么就用 SET CARRY ON。

具体写法是:

· SET CARRY ON　　　　（置到开状态）

· USE 人事↓

· *APPEND↓（自动将前一记录内容复制到下一记录中）*

……　　（不同的部分由人工进行修改）

系统的隐含状态是 *OFF*（关闭）。

3. SET COLON ON / OFF——是否用冒号定界

在使用 APPEND 和 EDIT 命令时已经看到，字段宽度是由冒号(:) 定出来的。如果我们希望取消冒号定界，那么可以在此之前设置:

·SET COLON OFF

表示 APPEND 和 EDIT 中不用冒号定界。系统隐含特征是 ON（要用冒号定界）。

4. SET CONFIRM OFF/ON——光标是否自动下跳

在用 APPEND EDIT INSERT CREATE 输入和修改数据库数据的过程中，当输入数据填满字段宽度时，如果希望敲↓键后才允许光标跳到下一个字段（或记录），那么在该命令使用之前必须设置：

·SET CONFIRM ON （取消光标自动下跳）

系统的标准状态是 OFF（填满字段宽度时光标自动下跳）

5. SET CONSOLE ON/OFF——是否输出到屏幕

在应用中，如果希望程序、命令、执行结果以及系统的提示信息等均不要在屏幕上显示出来，那么可以设置：

·SET CONSOLE OFF （屏幕不显示任何信息）

这通常用于键入加密口令等不允许人家看到自己的程序运行情况的时候。如果要恢复到屏幕显示，那么把 OFF 换成 ON。系统的标准状态是 ON。

6. SET DEBUG ON/OFF——是否将 *ECHO* 和 *STEP* 执行的结果送到打印机

为了分析和排除程序中的错误，有时需要将 *ECHO* 和 *STEP* 命令执行的结果传送到打印机上打印出来。可以设置：

·*SET DEBUG ON*

系统的标准状态是 *OFF*。*DEBUG ECHO STEP* 和 *TALK* 命令均用于程序的调试中。

7. SET ECHO ON/OFF——是否显示命令行

在调试程序中，如果我们希望在程序执行的同时，又在屏幕上显示出正在执行的程序，这样有利于程序的调试，那么可以在程序执行之前设置:

·SET ECHO ON

系统的标准状态是 OFF（不显示）。

8. SET EJECT ON/OFF——是否换页

如果打印完一张报表后需要换到另一页上再打印另一张报表，那么可以设置换页标志:

·SET EJECT ON

系统的标准状态是 ON（自动换页）。但需注意，这个命令对有些打印机可能不起作用。

9. SET ESCAPE ON/OFF——ESC中断是否有效

在 cdBASEⅡ中，不管处于何种情况，敲 Esc 键后系统就无条件回到点状态。如果要取消这一功能，那么设置：·SET ESCAPE OFF 这时，敲 ESC 键就不能回到点状态了。系统提供的标准功能是 ON。

10. SET EXACT ON/OFF——是否严格比较

所谓严格比较是指：若两个字符串比较相等，只有当两个字符串的字符和个数都对应相同时才称为相等。如：“ABC”=“ABC”是相等的，而“ABC”=“ABCDE”是不等的。

非严格比较是指：只要较短的字符串包含在较长的字符串之中且处于较长字符串的开头就行了。如：“ABC”=“ABC”是相等的，而“ABC”=“ABCDE”也是相等的。

系统的标准状态是 OFF（即非严格比较）。

11. SET INTENSITY ON/OFF——是否反相显示

在使用 APPEND, BROWSE 和 EDIT 命令中，字段名除了用冒号定界外，我们看到定界的部分更加光亮，以便衬托出需要的部分，这就是反相显示（即白底黑字）。这种显示方式可以控制，当不需要反相显示时，可以设置:

·SET INTENSITY OFF （取消反相显示）

系统的标准状态是 ON（需要反相显示）

12. SET LINKAGE ON/OFF——是否双区操作

如果我们希望把主区和辅区的记录联系起来，则可以设置双区联合操作。

·SET LINKAGE ON

这时当使用 LIST, DISPLAY, SUM 和 REPORT 命令时，主区和辅区两个数据库文件均起作用。而且字段个数可以放宽到 64 个，记录的字符个数也可以达到 2000 个字符。

系统的标准状态是 OFF（即主区和辅区彼此独立）。

13. SET PRINT ON/OFF——是否连通打印机

如果我们希望把操作过程中显示的内容，数据、程序以及提示信息在打印机上打印出来，那么可以事先连通打印机。

·SET PRINT ON （事先要安装好打印机并打开电源）

如果中华学习机上没有安装打印机，那么就不能用这条命令。

系统的标准状态是 OFF（不接通打印机）。

14. SET RAW OFF/ON——字段间是否留空格

如果希望 DISPLAY 或 LIST 显示的数据字段彼此间不留空格，那么可以设置：

·*SET　RAW　ON*↓（字段间不留空格）

·USE　商品↓

·DISPLAY↓　　　（字段值间无空格）

00001　115 自行车 170.93 环球-2

·SET　RAW　OFF↓（字段间要留空格）

·DISPLAY↓　　　（字段值间有空格）

00001　115　自行车　170.93　环球-2

系统的标准状态是 OFF（字段值彼此间要留空格）。

15. SET SCREEN ON / OFF——是否需要全屏幕操作

所谓全屏幕操作是指：在 CREATE、APPEND、EDIT、INSERT 和 BROWSE 命令的执行中，光标可以在屏幕中上下左右移动。如果希望关闭这种功能，那么可以设置状态为

·SET　SCREEN　OFF

系统的标准状态是 ON（需要全屏幕功能）。

16.　SET STEP　ON / OFF——是否暂停

在程序调试中，有时需要执行一条命令后就暂停，以便分析程序中的错误。这时可以设置状态为：

·SET　STEP　ON

当系统执行一条命令后暂停，并提示用户选择后继的操作：

Y：单步　　N：键盘命令　　Esc：退出

如果敲入字母 Y，表示继续一条一条命令的调试。

如果敲入字母 N，表示从键盘输入命令。

如果敲 Esc 键，表示退出命令文件的执行。

系统的标准状态为 OFF（即连续操作）

17. SET　TALK　ON / OFF——是否显示执行结果

如果我们不希望命令执行的结果显示在屏幕上，就可设置状态为：

```
·SET  TALK  OFF↓
·STORE  5*8  TO  X↓
·?  X↓
40
```

系统的标准状态为 ON（要显示命令执行的结果），因而第三章中很多命令执行的结果都显示出来了。

4.2.2　SET〈参数 1〉TO〈参数 2〉的用法

1. SET　FORMAT　TO　SCREEN / PRINT

将@命令执行的结果送到屏幕或打印机上输出。如果以后用@命令建立的报表不仅要在屏幕上显示出来，而且还要在打印纸上打印出来的话，那么一定要事先设置：

```
SET  FORMAT  TO  PRINT
```

没有配备打印机的系统不能用该命令。没有明确说明在打印机上输出时，系统自动在屏幕上显示出来。

2. SET　HEADING　TO　〈字符串〉

在输出用 REPORT 命令建立的报表时，可以在报表标题之前加上一个小标题。如设置：

```
·SET  HEADING  TO  "学习情况统计"
```

那么以后在调用报表文件打印报表时就自动在报表之前打印一个表头：学习情况统计。

3. SET　MARGIN　TO　〈n〉

设置打印纸左边所留的空白边。其中 n＜132。如：

SET MARGIN TO 10 表示打印纸左边空 10 个字符的位置后开始打印有效字符。

习题四

1. 如果要利用 INT 进行四舍五入取整，那么该如何办？请写出命令来。

2. @函数和 $ 函数有什么区别？举例说明。

3, STR 和 CHR 有什么区别？它们各自在哪种情况下使用？

4. ·STORE 5*8 TO X↓

   ·? &X↓

   40

   这是否正确？为什么

5. 请上机实习 SET CARRY ON 的功能。

6. 如果在 APPEND 中输入数据，我希望消除光标的自动下跳功能，应该设置什么状态？写出这个命令。

7. 请想一个实例，刚好希望字段间的显示不留空格。写出命令并显示出结果。

第五章　排序和索引

在第三章实习显示命令和定位命令时，如果记录个数较多，那么用户将明显地发觉，系统查找一个记录的速度太慢了．这是由于系统使用的查找方法引起的．以前使用的方法是一种顺序查找，即系统根据查找条件，一个记录一个记录依次比较，如果记录个数很多，而满足条件的记录又恰巧在数据库文件的尾部，那么花费的时间自然就长了．为了解决这个问题，cdBASEⅡ引入了另一种组织形式——排序和索引，使得记录的查找速度大大加快．

5.1　排序命令

所谓排序就是把数据库文件的记录按某些字段值递升或递降的次序重新排列的过程，从而生成一个有序的文件．

1.　排序命令

格式：SORT　ON　〈排序字段〉　TO　〈文件名〉
〔ASCEDING／DESCENDING〕

作用：将活化数据库文件的记录按指定字段值递升或递降的次序重新排列，从而生成一个新的数据库文件．

例 1. 将商品文件按单价由小到大排序．

```
·USE 商品↓
·SORT ON 单价 TO SSP↓
排序完成
·USE SSP↓
·LIST↓
00001  267  电饭锅     38.42  快熟牌
00002  148  电热褥     88.77  助温牌
00003  158  缝纫机    161.26  家佳乐
00004  115  自行车    170.93  环球-2
00005  127  收录机    679.95  神笛牌
00006  145  电冰箱    979.41  北极星
00007  333  电视机   1083.00  美乐-9
```

例 2. 按品名中字母降序来排列各种商品（字母降序是：Z Y X……C B A，升序是：A B C……X Y Z）。

```
·USE 商品↓
·SORT ON 品名 TO SPM DESCENDING↓
排序完成
·USE SPM↓
·LIST↓
00001  115  自行车  170.93    环球-2
00002  127  收录机  679.95    神笛牌
00003  158  缝纫机  161.26    家佳乐
00004  338  电视机  1083.00   美乐-9
00005  149  电热褥  88.77     助温牌
00006  267  电饭锅  38.42     快熟牌
00007  145  电冰箱  979.41    北极星
```

注意：

①排序后生成的新文件不能是排序文件本身。

例如：我们将商品文件排序，那么短语"TO"之后的文件名一定不能又是"商品"。排序并不影响待排序文件（商品）。

② ASCENDING 表示递升（由小到大），DESCENDING 表示递降（由大到小）。如果都省略，那么隐含按递升排序。汉字的升序或降序实际是按拼音字母顺序。

③排序后生成的新文件仍然是数据库文件。

2. 对排序新文件的操作

由于排序后生成的新文件仍然是一个数据库文件，因而，凡是对数据库文件能够进行的操作均可用于排序后的新文件。

5.2 索引文件及其操作

索引是不同于排序的又一种文件组织形式。其组织形式和作用类似于一本书的目录。当我们要查找一本书的内容时，经验告诉我们：先查看目录，根据目录找到页码，再根据页码找到具体的内容。从查找速度来说，这种方式肯定是最快的。现在的问题是要先建立目录，这就是索引命令的作用。

5.2.1 索引文件的建立

索引文件是通过 INDEX 命令来建立的。

格式：INDEX ON 〈索引字段表达式〉 TO 〈文件名〉

作用：在活化数据库文件的基础上，按给定的索引字段

值递升的次序建立一个新的有序文件。

例 3. 按品名来建立一个索引文件.

```
·USE  商品↓
·INDEX  ON  品名  TO  ISP↓
00007 条记录被索引
(打开索引文件 ISP)
·USE  商品  INDEX  ISP↓
·LIST↓
00002  145  电冰箱   979.41      北极星
00003  267  电饭锅    38.42      快熟牌
00007  149  电热褥    88.77      助温牌
00004  333  电视机  1083.00      美乐-9
00006  158  缝纫机   161.26      家佳乐
00005  127  收录机   679.95      神笛牌
00001  115  自行车   170.93      环球-2
```

例 4. 按两个相同类型的字段来建立索引. 假设有一个销售文件除缝纫机外，其它商品均恢复原价，其结构和数据如下:

记录号	姓 名	品名	单价	数量	金额
00001	张森林	电冰箱	*1030.96*	*2*	*2061.92*
00002	张森林	收录机	*715.74*	*1*	*715.74*
00003	徐荣策	自行车	*179.93*	*5*	*899.65*
00004	徐荣策	电视机	*1140.00*	*3*	*3420.00*
00005	吕速才	电冰箱	*1030.96*	*2*	*2061.92*
00006	吕速才	收录机	*715.74*	*3*	*2147.72*

要求按品名索引，如果品名相同，再按售货员姓名索引。

· *USE* 销售↓

· *INDEX ON* 品名+姓名 *TO IXS*↓

00006 条记录被索引

· *USE* 销售 *INDEX IXS*↓ (打开索引文件)

· *LIST*↓

00005	吕速才	电冰箱	*1030.96*	*2*	*2061.88*
00001	张森林	电冰箱	*1030.96*	*2*	*2061.92*
00004	徐荣策	电视机	*1140.00*	*3*	*3420.00*
00006	吕速才	收录机	*715.74*	*3*	*2147.22*
00002	张森林	收录机	*715.74*	*1*	*715.74*
00003	徐荣策	自行车	*179.93*	*5*	*899.65*

例 5. 按两个不同类型字段来建立索引. 我们仍然以销售文件为例. 先按售货员姓名索引, 姓名相同再按单价索引.

· *USE* 销售↓

· *INDEX ON* 姓名+*STR(*单价, *7,2)TO IXD*↓

00006 条记录被索引

· *USE* 销售 *INDEX IXD*↓

· *LIST*↓

00005	吕速才	电冰箱	*1030.96*	*2*	*2061.92*
00006	吕速才	收录机	*715.74*	*3*	*2147.72*
00004	徐荣策	电视机	*1140.00*	*3*	*3420.00*
00003	徐荣策	自行车	*179.93*	*5*	*899.65*
00001	张森林	电冰箱	*1030.96*	*2*	*2061.92*
00002	张森林	收录机	*715.74*	*1*	*715.74*

在具体实现索引时, 系统需要建立两个区 (不是前面所说的工作区); 一个是存放数据的, 称为数据区 (将就数据库文件来充当); 另一个是索引字段的值和记录号的一个对

照表，称为索引区。建立索引的过程就是生成索引区。

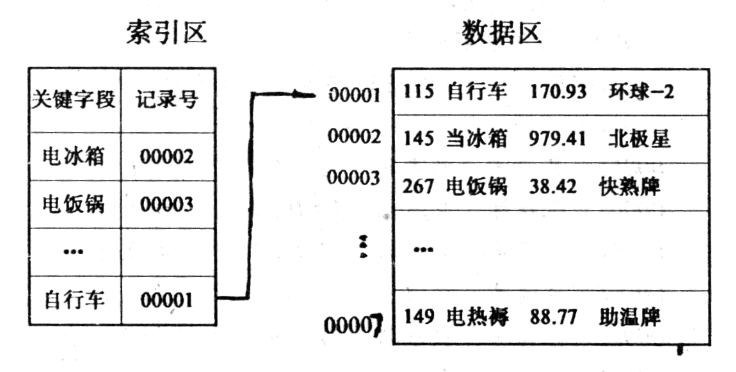


图 5.1　索引示意图

当给定一个索引字段值来检索一个记录时，系统首先在索引区内查找。因为索引区只有两项，相对来说比较小，而且索引字段值是有序的，因而很快就能查到给定的索引字段值（当然该值对应的记录要存在于数据库中），按照对应的记录号，马上就可以把记录取出来。因而查找速度是相当地快。从这里也可以看出，索引区是依赖于数据库文件的，没有数据库文件，索引区内指定的记录就无法找到，因为索引区内并不实际存放记录。

使用中需要注意以下几点：

①索引字段表达式可以是用"+"连接起来的多个字段名；如果其中有数值型字段，则必须用 STR 函数转换成字符型。参见例 4 和例 5。

②最重要的索引字段必须先写。如：品名+姓名。

③索引总是递升的。索引文件的扩展名是·NDX。

④逻辑型字段不能作索引字段。

⑤索引字段表达式中字符个数<100个。

⑥同一数据库文件上至多可以建立7个不同的索引文件。

5.2.2 对索引文件的操作

建立索引文件的目的是为了快速地查找数据库的记录。

1. 打开索引文件

格式：USE〈数据库文件名〉INDEX〈索引文件名字表〉

作用：打开基于数据库文件的索引文件。

其中：索引文件名字表是基于数据库文件上建立的一个或多个索引文件。如果是多个索引文件，彼此间要用逗号隔开。

例6.　　·USE 销售 INDEX IXS, IXD↓

这就打开了建立在销售文件上的IXS和IXD索引文件。排在索引文件名字表中的第一个索引文件（IXS）称为主索引文件。当用USE打开索引文件时，指针指向主索引文件的第一个记录（索引区的第一个记录，不一定是数据库文件的第一个记录），随着命令的执行，指针可以来回在主索引区上拔动。

注意：

①一个USE一次可以打开建立在同一数据库文件上的至多7个索引文件。

②若使用GOTO SKIP命令，那么定位的是索引的位置，而不是数据库文件的位置。

③如果索引文件是打开的，那么用APPEND、EDIT、REPLACE和PACK命令引起的变化将自动反映到索引中来。

2. 查找命令 FIND

格式：FIND　〈字符序列〉

作用：查找索引文件中索引字段表达式的值同给定的字

符序列相匹配的第一个记录。

例 7. · *USE 销售 INDEX IXD*↓

· *FIND 张森林 715.74*↓

· *DISPLAY*↓

00002 张森林 收录机 715.74 1 715.74

注意:

①字符序列不能带引号。如：FIND“张森林 715.74”是错误的。

②若索引字段值包含了若干个前导空格（即值的左边有若干个空格），则字符序列也必须包含相同个数的空格且用引号将空格括起来。

③字符序列必须是索引字段表达式值的全部或开头连续的若干个字符才有效，不能从值中间抽出部分作为字符序列。如：FIND 森林或 FIND 715 都是不正确的。而 FIND 张或 FIND 张森或 FIND 张森林是可以的。

④当字符序列是一个内存变量的值时，必须用宏替换&。如:

例 8. · *STORE “张森林” TO W*↓

· *FIND &W*↓

· *DISPLAY*↓

00002 张森林 收录机 715.74 1 715.74

⑤FIND 只能定位满足条件的第一个记录。

5.3 对有序文件的操作

我们把排序后得到的新文件和索引文件统称为有序文件。把用于排序和索引的字段称为关键字段。对有序文件，

cdBASEⅡ提供一些专门的命令，下面介绍之。

1. 更新命令 UPDATE

如果我们希望用一个文件的内容来更新另一个文件的内容时，就可用更新命令。

格式：UPDATE FROM 〈文件名〉ON 〈关键字段〉〔ADD〈字段表〉〕〔REPLACE〈字段表〉〕

作用：用指定文件的内容来修改活化数据库文件的相应字段内容。

其中：关键字段用于构成修改的条件。即是说：如果两个文件（指定的文件和活化数据库文件）在关键字段上的值相等，那么就可以进行修改。ADD 和 REPLACE 指出了修改的方式。ADD 表示累加，REPLACE 表示替换。

我们还是先来看一个例子。

例 9. 假设库存文件和货单文件都是有序文件，它们的关键字段是品名（即按品名排序）。

库存文件

记录号	品名	数量
00001	电冰箱	55
00002	电饭锅	231
00003	电热褥	233
00004	电视机	150
00005	缝纫机	144
00006	收录机	23
00007	自行车	38

货单文件

记录号	品名	数量
00001	电冰箱	158
00002	收录机	166
00003	自行车	120

现在需要把新货入库，即把货单中商品的数量值累加到库存文件的同类商品数量值中。

· SELECT SECODARY ↓

· *USE　货单* ↓

· *SELECT　PRIMARY* ↓

· *USE　库存* ↓

· *UPDATE FROM 货单 ON 品名 ADD 数量* ↓

· *LIST* ↓

00001　电冰箱　213

00002　电饭锅　231

00003　电热褥　233

00004　电视机　150

00005　缝纫机　144

00006　收录机　189

00007　自行车　158

通过执行 UPDATE 命令就把货单文件的数量累加到库存文件的相应数量值上去了.

UPDATE 的执行情况如下图所示:

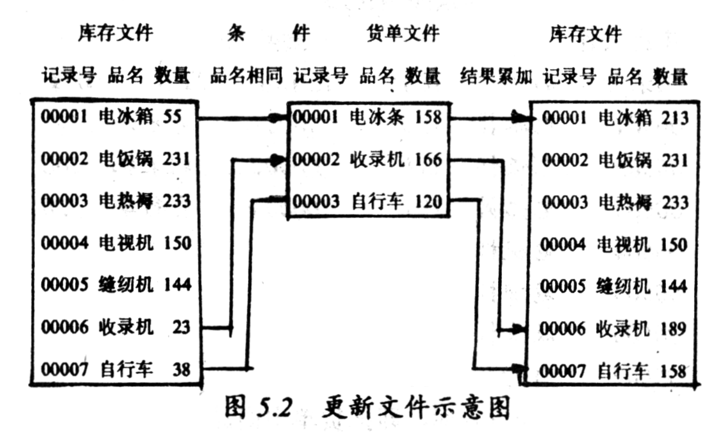


图 5.2　更新文件示意图

系统执行 UPDATE 命令的过程是这样的:

按库存文件的记录顺序，用品名的值同货单文件的对应字段（品名）的值进行比较。如果字段值相等，就把货单文件的指定字段值累加到库存文件的数量值上，然后推到库存文件的下一个记录，重复进行比较、累加。如果比较后发现值（品名的值）不等，库存文件自动跳到下一个记录，再重复比较和累加的过程，直到库存文件记录完毕为止。

使用时需要注意：

①两个文件都必须按关键字段排序或索引。

②只有数值型字段才能用累加（ADD）方式。

③当关键字段相同的记录多于一个（即关键字段值对应的记录不唯一）时，只对第一记录进行累加或替换。

④如果同时累加（ADD）或替换（REPLACE）多个字段，则各字段间要用逗号隔开。

2. 合并细项 TOTAL

在实际应用中，经常要进行如象流水帐合计之类的工作，这可用合并细项命令来实现。

格式： TOTAL ON 〈关键字段〉 TO 〈文件名〉

〔〈范围〉〕〔FOR〈条件〉〕〔FIELD〈字段表〉〕

作用： 对活化有序文件按关键字段值分组求和。

其中： FIELD〈字段表〉指出了需要合并细项的字段名。我们还是用例子来说明这个命令的用法。

例 10. 假设我们需要按售货员的姓名来统计每个人的销售金额，实际就是将各人销售商品的金额加起来。（假定销售文件按姓名索引后为销售Ⅰ）

· USE 销售 INDEX 销售Ⅰ↓

· TOTAL ON 姓名 TO TFIILE FIELD 金额 ALL↓

```
00003   条记录被拷贝
·USE  TFILE↓
·COPY TO 营业额 FIELD 姓名，金额↓
00003  条记录被拷贝
·USE  营业额↓
·LIST↓
00001  吕速才  4209.14
00002  徐荣策  4319.65
00003  张森林  2777.66
```

TOTAL 命令的执行情况如下所示：

记录号	姓名	品名	单价	数量	金额	记录号	姓名	金额
00001	吕速才	电视机	1030.96	2	2061.92	00001	吕速才	4209.14
00002	吕速才	收录机	715.74	3	2147.22			
00003	徐荣策	自行车	179.93	5	899.65	00002	徐荣策	4319.65
00004	徐荣策	电视机	1140.00	3	3420.00			
00005	张森林	电冰箱	1030.96	2	2061.92	00003	张森林	2777.66
00006	张森林	收录机	715.74	1	715.74			

由此可以看到：TOTAL 命令实际就是将按姓名分组的记录的金额值加起来就行了。

需要注意的是：

①如果不用范围和条件，那么对所有记录合并细项。

②如果不用 FIELD 指出合并的字段，那么将对所有数值型字段合并细项。

③合并细项后得到的新文件同待合并细项的老文件有相同的结构。新文件除了合并项（例如金额字段）的值是新值以外，其它值如：姓名、品名、单价、数量均是旧文件相应分组的头个记录值。

④由于新旧文件有相同的结构，如果新文件的字段宽度(如：金额字段的宽度小于7）不足以容纳合并值，则系统在合并字段（金额）中放置星号*，作为数据溢出标志。因而，凡是要进行分组合并的数据库文件，在定义其合并字段宽度时应适当加大。

习题五

1. 举出一个需要用到排序和索引的例子，并分别用SORT和INDEX进行排序和索引。

2. SORT和INDEX的区别是什么？

3. 如果一个数据库文件要按多个字段排序，如：学生高考成绩文件，首先按总分由高到低排序，总分相同再按数学、物理、化学、语文等分数由高到低排序，请写出实现这种排序的语句。

4. USE〈数据库文件〉INDEX〈索引文件名字表〉这个命令虽然可以打开多个索引文件，而指针却只能作用于主索引文件。那么，这时打开多个索引文件还有什么意义？请举一个实例来说明，最好写出语句。

5. 对索引文件能否用LOCATE来查找（定位)？对排序文件能否用FIND进行检索（定位)？

6. 对索引文件要增加或减少一些记录怎么办？用语句来完成。

7. 请举一个例子，它适宜于用UPDATE命令来更新且必须用到替换（REPLACE)。写出语句。

8. 分别建立销售和商品两个数据库文件并装入适当数据。文件结构是：

销售（姓名、品名、单价、数量、金额），商品（品

名、型号、产地）写出求解下列问题的命令。

①找出销售了张三所售的某一种商品的其他售货员的名字，

②找出畅销商品的名称和产地。所谓畅销商品是指销量在1000件以上且累计金额不低于20万元的商品。

第六章　命令文件和报表格式设计

前几章介绍了 cdBASEⅡ命令在对话方式下的用法。显然，学起来容易，用起来简便。缺点是速度太慢，自动化程度太低，这就直接影响到 cdBASEⅡ的实际应用价值。为解决这一问题，系统又提供了程序的工作方式，或称为 cdBASEⅡ的程序设计，这是通过命令文件来实现的。

数据库技术最适宜于进行数据处理。在数据处理中经常要用到各种报表，cdBASEⅡ也提供了设计实用报表的能力和方法。可以说，最有用的报表格式也是通过命令文件的形式表现出来的。因此，本章我们先介绍命令文件。

6.1　命令文件

所谓命令文件就是根据实际应用要求，一定规则和顺序排列的 cdBASEⅡ命令序列。它以文件的形式保存在磁盘上，以后可以用专门命令来执行。命令文件也叫做程序，以后可以不加区别地使用这两个概念。命令文件的最大好处是可以使我们的操作自动化，程序规范化。

6.1.1 命令文件的建立方式

从理论上来说，命令文件的建立方法是简单的。

格式：MODIFY COMMAND 〈文件名〉

〈命令序列〉

RETURN

作用：允许用对话的方式建立和修改一个命令文件。

其中：〈文件名〉就是命令文件的名字，它的扩展名是·COM。命令文件建立成功以后，磁盘上保存的就是这个名字。

〈命令序列〉就是完成某种或某些操作而要用到的一个或多个命令。这些命令要按有关规则和顺序写出，一个命令占一行，每个命令之后用↓换行。

RETURN 可以看成是命令文件的结束标志，在子程序中是必要的。

我们来看一个命令文件实例。

例 1 在第五章例 10 中，要求按售货员的姓名来统计每个人营业额。那里是用对话方式来实现的。现在用命令文件写出就是：

·*MODIFY COMMAND MCFI*↓(命令文件名是 MCFI)

新文件（系统提示，表明建立新文件，以下用户敲入）

USE 销售↓

INDEX ON 姓名 *TO* 销售Ⅰ↓

USE 销售 *INDEX* 销售Ⅰ↓

TOTAL ON 姓名 *TO TFILE FIELD* 全额 *ALL*↓

USE TFILE↓

COPY TO 营业额 *FIELD* 姓名，全额↓

USE 营业额↓

LIST↓

RETURN　　　　　　（敲∧W 将命令文件存盘）

至此，一个命令文件就建好了。建立过程是这样的：

①首先在点状态下敲入建立命令文件的标志：

MODIFY　COMMAND　以及文件名 MCFI。

②屏幕上用"新文件"作为提示，意思是用户现在建立的是一个新文件。屏幕空出，等待用户敲入命令。

③从上到下，从左到右依次敲入各条命令，每条命令输入完毕要敲↓键。从上面可以看到，命令之前没有小圆点了，这是程序和对话工作方式的差异之处。

④全部命令输入完毕，敲∧W，其作用就是告诉系统，现在需要把输入的内容保存在磁盘上。敲∧W 存盘后又回到点状态。这时我们可以进行其它操作。

建立命令文件时需要注意：

①一个命令占一行。如果一个命令较长，一行容纳不下，可以由系统自动换行（不用敲↓），或在适当的位置敲入一个分号；表示后继行为续行。

②每一行输完以后一定要敲↓键，表示该行结束。为简洁，以后程序行末尾我们不再给出↓，但用户输入命令时是不能少掉的。切记！

③整个命令文件输入完毕，一定要敲∧W 记盘。如果是大型程序，为防止机器故障而前功尽弃，最好输入一部分就敲∧W 记盘后再接着输入。

④经验告诉我们：用 CREATE 建立数据库文件的操作最好在点状态用对话方式来完成。而对文件的检索等操作最好用命令文件的形式来完成。

6.1.2 命令文件的执行

从上例已经看到，在敲入命令后，系统并不立刻执行。这是程序和对话工作方式的又一不同之处。命令文件建好以后，必须用专门的执行命令来运行。

1. 执行命令 DO

格式：DO 〈命令文件名〉

作用：打开并执行指定的命令文件。

例 2. 调用例 1 中建立的命令文件 MCFI。

·DO MCFI↓ （以下是运行提示和结果）

00006 条记录被索引

00001 吕速才 4209.14

00002 徐荣策 4319.65

00003 张森林 2777.66

·

由于是在点状态下调用，因而运行结束后仍然回到点状态。

2. 命令文件的多重调用

由命令文件的建立和执行，我们自然会联想到：既然命令文件是由语句（语句和命令两个术语经常等同使用）序列组成的，那么这些语句序列中能否包含 DO 语句呢？即一个命令文件中又调用另一个命令文件。回答是肯定的。由此，引出了命令文件的多重调用问题。

有两个概念需要事先解释一下。

主程序和子程序——如果甲程序（命令文件和程序也等同使用）中调用了乙程序，那么称甲程序为主程序，而乙程序称为子程序。主程序和子程序是相对的。在某种场合是子

程序；在另一种场合（如乙调用丙）又可能是主程序。这如同几代同堂的家族中的老子儿子的关系一样。但通常的习惯却是把不被调用的程序称为主程序。凡是被其它程序调用的程序都叫做子程序。主程序和子程序的调用关系如下：

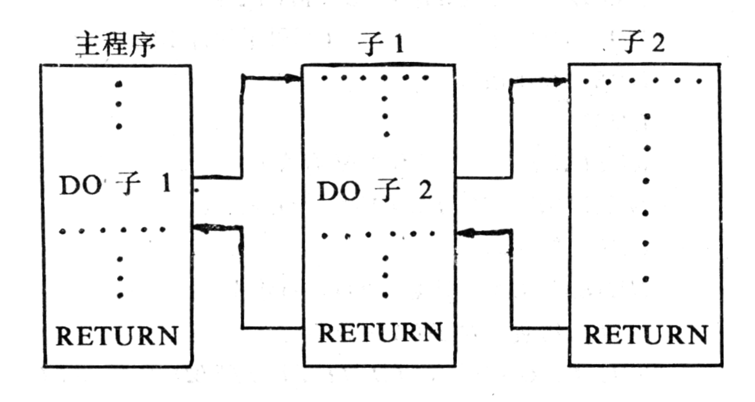


图 *6.1* 主子程序的调用示意图

调用过程是：主程序执行到“DO 子 1”时，就调用“子 1”程序，相当于转到“子 1”程序的第一条命令。当“子 1”执行到“DO 子 2”时，就转到“子 2”的入口，继续执行。当遇到“子 2”的“RETURN”语句时，控制返回到“子 1”中“DO 子 2”的下一条语句，再继续执行。当遇到“子 1”中的“RETURN”时，又自动转到“主程序”中“DO 子 1”的下一条命令继续执行，直到主程序结束为止。整个调用过程遵循的是“哪来哪去”的原则，即调用了某个子程序，执行完后还得回到调用处的下一个语句。这样形成一个闭合回路。

从理论上来说，这种一重一重的嵌套调用关系还可以一直延续下去。但由于具体硬件的限制，最多可以嵌套 16

层。一个程序中可以多次调用一个子程序，当然也可以调用多个子程序，但总的调用层次不能超过 16 层。因为 cdBASEⅡ规定同时打开的文件个数不能超过 16 个。

例 2. 用主子程序的形式来改写例 1。

```
·MODIFY  COMMAND  MCF2↓
 DO  MCF11            (调用子程序 MCF11)
 USE  营业额
 LIST
 RETURN        (敲∧W 存盘)
·MODIFY  COMMAND  MCF11↓
 USE  销售
 INDEX  ON  姓名  TO  销售 I
 USE  销售  INDEX  销售 I
 TOTAL ON 姓名 TO TFILE FIELD 金额 ALL
 USE  TFILE
 COPY  TO  营业额  FIELD  姓名，金额
 RETURN   (∧W)
·DO MCF2↓      (执行主程序，结果同例 1)
```

这个例子说明主程序 MCF2 中调用了子程序 MCF11。

例 3. 我们也可以把 MCF11 子程序改写成主子程序的形式。

```
·MODIFY  COMMAND  MCF111↓ (子程序)
 USE  销售
 INDEX  ON  姓名  TO  销售 I
 DO  MCF112
 USE  TFILE
```

```
COPY TO 营业额 FIELD 姓名，金额
RETURN (∧W)
·MODIFY COMMAND MCF112↓ (子程序)
USE 销售 INDEX 销售I
TOTAL ON 姓名 TO TFILE FIELD 金额 ALL
RETURN (∧W)
·MODIFY COMMAND MCF3↓ (主程序)
DO MCF111
USE 营业额
LIST
RETURN (∧W)
·DO MCF3↓       (运行主程序)
```

执行过程是：MCF3中调用MCF111，MCF111又调用MCF112。我们可以把调用简单地理解为将子程序搬到调用语句的位置来执行。由上可以看到，子程序也是用命令文件来实现的。子程序为编写和调试大型程序提供了有效的处理方法。在实际应用中，我们可以将一个大型问题分为若干块，每一块用一个子程序来实现。当然，子程序中还可以再套子程序。一般说来，一种相对独立的操作（可能要用多个命令才能实现）就可作成一个子程序。然后用一个总控程序将各个子程序串起来，组成解决实际问题的一个应用程序(参见后面的应用实例程序)。

在编写和调用子程序时注意：①子程序中有一个RETURN（返回）语句。②除了用DO来调用程序以外，也可以在操作系统下调用，格式为：A>DBS〈文件名〉，这时系统启动cdBASEⅡ后自动执行命令文件，运行完毕回到操作系统状态。

6.1.3 命令文件的修改

命令文件的修改包括：添加、减少或更改命令以及因敲键失误而造成的字符错误。无论哪种情况，都可用控制键将光标移到出错处，再更正之。

MODIFY COMMAND 中允许使用的光标控制键有：

∧*E*——光标上移一行。

∧*X*——光标下移一行。↓具有相同功能。

∧*S*——光标左移一个字符。有◁同种功能。

∧*D*——光标右移一个字符。

∧*T*——删去光标所在行，后面的行依次递推上移。

∧*N*——将光标所在行及后面所有行依次下推，留出一个空白行，以便插入新行。

∧*G*——删去光标所在字符。

∧*V*——允许在光标左边插入字符。

∧*W*——文件存盘后回到点状态。

∧*Q*——文件不存盘回到点状态。

例 4 修改命令文件 MCF3，要求在显示营业额之前加一个提示信息。

```
·MODIFY  COMMAND  MCF3↓
 DO  MCF111
 USE  营业额
 LIST
 RETURN
```

现在希望在 USE 营业额和 LIST 之间插入一个新行。方法是：当我们敲 MODIFY COMMAND MCF3↓以

后，屏幕显示出原来输入的命令，光标处在第一个命令行“DO MCF111”的第一个字符位置。敲↓或∧X 使光标移到“LIST”上，再敲∧N。这时屏幕上空出一行（LIST 自动下推），我们就敲入添加的命令。屏幕显示如下：

DO　MCF111

USE 营业额

?“下面开始显示数据”（这是新插入的命令行）

LIST

RETURN

同时还可以修改发生的其它错误，敲∧W 就把修改后的内容存盘（代替原文件的内容）。如果只希望在屏幕上修改而不改变磁盘原来文件的内容，那就不敲∧W 而敲∧Q。∧Q 的作用是撤消修改后回到点状态。

注意：命令文件建立和修改方式相同，即建立和修改都用 MODIFY COMMAND 〈文件名〉。那么系统如何判断现在是要建立新文件呢还是修改旧文件？如果文件名是磁盘上已经存在的命令文件名，系统就认为是要进行修改，因而显示命令行。否则就是建立新文件，因而给出空白屏幕。

6.2 专用于程序的命令

程序中除了可以用对话方式下那些命令以外，还有一些专用的命令。所谓专用是指离开了程序，这些命令就没有什么意义了。再者，个别命令的完整形式本身就是一段程序。

6.2.1　键盘输入命令

为了增强与程序的交互作用，使得在运行中可以方便地

干予程序的执行，系统提供了一组键盘输入命令。

1. 等待命令 WAIT

格式： WAIT〔〈提示〉〕〔TO〈内存变量〉〕

作用： 暂停程序的执行，等待从键盘上敲入任何一个字符后又继续后继命令的执行。

其中：〈提示〉是用户自己根据需要给出的，键盘上敲入的值可以保存在内存变量中。

例 5. *WAIT "请输入数据" TO W*

执行该命令后屏幕提示：

请输入数据：

注意：

该命令只能接收一个字符；而且，不论我们敲入的是一个字母还是数字，都作为字符来接收。一般的 dBASEⅡ不具备〔〈提示〉〕功能。

2. 接收命令 ACCEPT

格式： ACCEPT〔〈提示〉〕TO〈内存变量〉

作用： 允许从键盘上接收一串字符并送到内存变量中保存起来。

例 6. ACCEPT "请输入姓名" TO NAME

执行该命令后，屏幕提示"请输入姓名"并等待我们从键盘上敲入姓名值。

请输入数据：张三↓

系统自动将"张三"保存到内存变量 NAME 中。

注意：ACCEPT 允许接收一个或多个字符并作为字符型数据保存在内存变量中；输入字符之后一定要用↓作为结束标志；输入字符也不能带引号。

3. 输入命令 INPUT

格式：INPUT〔〈提示〉〕TO〈内存变量〉

作用：允许从键盘上接收一个表达式的值到内存变量中。

例 7. INPUT“输入数据”TO H

执行该命令后屏幕显示：

输入数据：*5+3 * 8*↓

系统将 5+3 * 8 的值 29 送到 H 中。

注意：INPUT 接收的是表达式的值。表达式可以是算术、关系、逻辑或字符串表达式。

4. 读入命令 READ

格式：@〈行，列〉SAY〈表达式〉GET〈变量名〉

…

READ

作用：允许从键盘上读入一个值到变量中。

这是一种格式输入输出命令，其中的〈行，列〉是指屏幕的座标位置。SAY〈表达式〉表示输出，GET〈变量〉表示输入。

例 8. 用屏幕格式输入输出来修改商品文件的第 3 号记录。

```
· MODIFY  COMMAND  MCF8↓
    USE  商品
    GO  3
    ERASE
    @ 2, 10  SAY“修改单价”GET  单价
    @ 3, 10  SAY“修改型号”GET  型号
    READ
    @ 4, 8  SAY“修改后的记录如下:”
    DISPLAY
```

RETURN

·DO　MCF8↓

修改单价：34.89

修改型号：乐天牌

修改后的记录如下：

267　电饭锅　34.89　乐天牌

程序的执行是这样的：机器在屏幕的第2行第10列显示“修改单价”，第3行第10列显示“修改型号”。光标处在单价的冒号之后，这时敲入：“34.89”以及“乐天牌”，系统将值分别送到单价和型号字段中。这两个值是通过READ来得到的。需要注意的是：

①GET和READ必须联用。如果GET之后是内存变量名，那么必须事先赋初值。

②利用这种命令，我们可以构造一个直观的输入输出格式。

6.2.2　条件语句

在应用中经常要根据条件来选择执行某些操作动作，这就是条件语句的作用。

1.　如果语句 IF

格式：IF 〈条件〉

```
        〈命令序列1〉
  〔ELSE
        〈命令序列2〉〕
  ENDIF
```

作用：如果条件成立，则执行命令序列1，否则执行命令序列2，然后转到ENDIF的后继命令去执行。

例9.　如果W的值是Y那么就打开商品文件．然后显

示之。否则就打开销售文件，再显示之。

```
·MODIFY  COMMAND  MCF9↓
WAIT"请输入数据:" TO  W
IF  W="Y"
USE  商品
LIST
ELSE
USE  销售
LIST
ENDIF
?"操作结束"
RETURN   (∧W)
·
```

显然，IF……ELSE……ENDIF 用于两个分支选择是非常方便的。其执行流程如下:

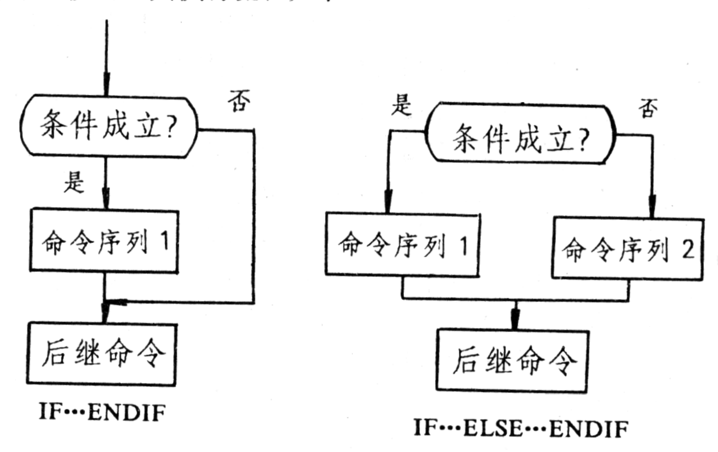


图 6.2 如果语句执行流程

2. 情形语句 *CASE*

如果应用中要进行两个分支以上的选择，那么用 DO　CASE 或许更为方便.

格式：DO　CASE

　　　　CASE〈条件 1〉

　　　　〈命令序列 1〉

　　　　CASE〈条件 2〉

　　　　〈命令序列 2〉

　　　　…

　　　　〔OTHERWISE

　　　　〈命令序列 n〉〕

　　ENDCASE

作用：构成多分支选择. 如果条件 1 成立，则执行相应的命令序列 1. 否则，依次检查条件，哪一个条件成立就执行相应的命令序列. 如果给出的所有条件均不成立而又带有 OTHERWISE 项，那么就执行命令序列 n. 相应命令序列执行后均转到 ENDCASE 的后继命令去执行.

例 10. 假设需要在不同的情况下显示不同文件的记录，我们可以用 CASE 命令来实现这种选择.

```
· MODIFY  COMMAND  MCF10↓
  ?  "这是一个选择不同文件的程序"
  ACCEPT "请输入文件名:" TO  FNAME
  DO  CASE
  CASE  FNAME="商品"
  USE  商品
  LIST  FOR  "电"$品名
  CASE  FNMAE="销售"
```

```
USE 销售
LIST FOR 数量>2
CASE FNAME="营业额"
USE 营业额
LIST FOR 金额>4000
OTHERWISE
? "没有这个文件"
ENDCASE
RETURE (∧W)
```

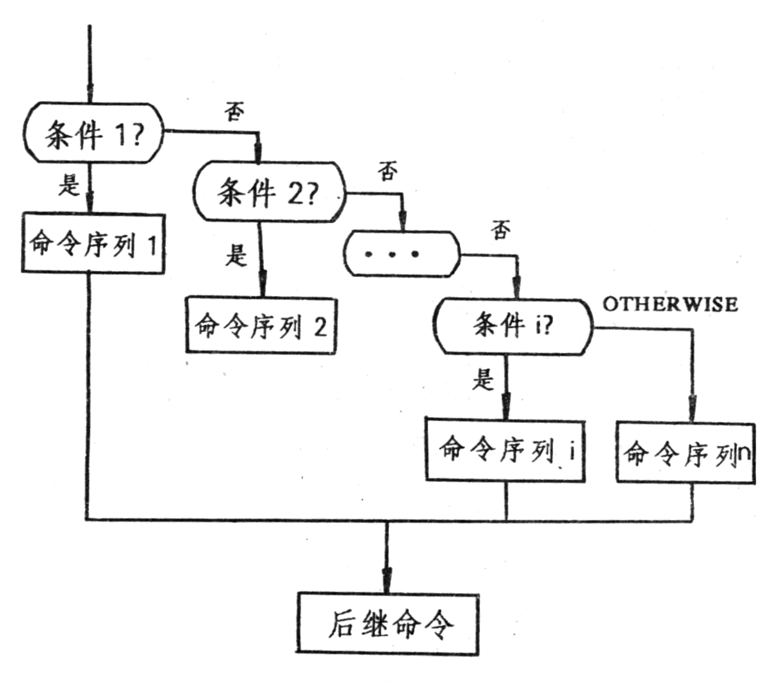


图 6.3 *CASE* 执行流程图

注意：在所有分支情形中，一旦有一个条件为真，就执行相应的命令序列。如果后面的条件也为真，系统也不会再来执行。只有前面条件不成立时才会进行下一个选择。其执行流程如图 6.3

6.2.3　循环

重复执行某些操作，但每次重复却有不同的内容，这种情况可以用循环恰当地描述出来。

1.　循环语句 DO WHILE

格式：DO WHILE 〈条件〉

〈命令序列〉

ENDDO

作用：当条件为真时，反复执行指定的命令序列，直到条件为假后就转到 ENDDO 的后继命令去执行。

其中：〈条件〉可以是一个关系表达式或逻辑表达式，它的值决定了是否循环。如果其值为真，则循环，否则就不循环而转到 ENDDO 之后。

〈命令序列〉又称为循环体，它是需要循环执行的部分。通常由一个或多个命令组成。

循环语句的执行过程是：

①计算条件的值，判断其是否为真。

②若条件值为真，执行循环体后又转第一步。

③若条件值为假，跳出循环转到 ENDDO 的后继命令去执行。

例 11　假设有一个工资文件，其结构如下：

工资（姓名、基本工资、奖金、福利费、房租费、水电费、实发工资）现要求把基本工资<70 元的增加 10 元，其余增加 8 元。

```
·MODIFY  COMMAND  MCF11
    USE  工资
    DO  WHILE  NOT.EOF
    IF基本工资<70
    REPLACE基本工资 WITH 基本工资+10,
    实发工资 WITH 实发工资+10
    ELSE
    REPLACE基本工资 WITH 基本工资+8
    实发工资 WITH 实发工资+8
    ENDIF
    SKIP
    ENDDO
    LIST
    RETURN    (∧W)
    ·DO  MCF11↓
```

运行结果略去了。循环语句的执行流程如图 6.4 所示。

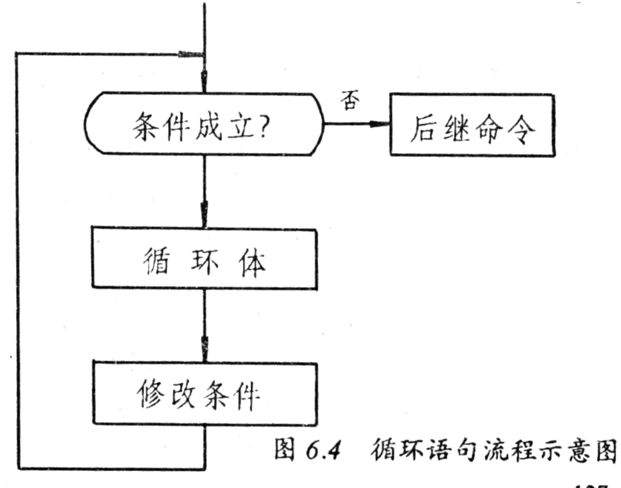


图 6.4　循环语句流程示意图

2. 回路语句 LOOP

格式：LOOP

作用：无条件转到循环的开头，准备进行下一次循环。

例 12. 找出一次销售金额就超过 3000 元的售货员姓名、品名及其金额。

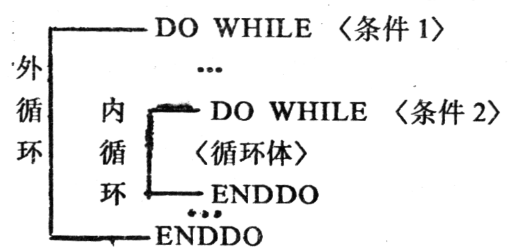
```
·MODIFY COMMAND MCF12↓
  USE 销售
  DO  WHILE  .NOT.EOF
  IF 金额<=3000
  SKIP
  LOOP
  ENDIF
  DISPLAY 姓名、品名、金额
  SKIP
  ENDDO
  RETURN   (∧W)
  ·DO  MCF12↓
  00004  徐荣策  电视机  3420.00
  ·
```

程序的执行过程是：打开销售文件，记录指针指向第一个记录。判断循环语句条件.NOT.EOF 的值是否为真。由于记录没有结束，所以 EOF 的值为假，.NOT.EOF 表示非假即为真。因此得出循示条件.NOT.EOF 的值为真，所以执行循环。如果金额值小于或等于 3000，那么就把记录的指针下推后执行 LOOP 而无条件进行下次循环。如果金额值大于 3000，就跳到 ENDIF 之后，显示满足要求的记录后再检查下一个记录。这样直到文件结束为止。

注意：**LOOP** 通常和 **IF** 语句联用以跳过某些语句。

3. 多重循环

如果循环语句的循环体中又包含有循环语句，这就构成所谓的多重循环。多重循环的一般形式如下：



从这种嵌套结构的二重循环可以看到，外循环执行一次，内循环就要执行多次。

例 13. 假设工资文件中增加了一项单位字段，表示某人是属于哪个部门的。完整结构如下：

工资（姓名、单位、基本工资、奖金、福利费、房租费、水电费、实发工资）现要求按单位统计实发工资（即将同一个单位职工实发工资累加起来）。

解决该问题的一种方法是先按单位排序文件，然后再累加同单位职工的实发工资。这个程序如下：

```
· MODIFY COMMAND MCF13↓
    USE  工资
    SORT ON 单位 TO GZS (按单位排序)
    USE GZS
    STORE 单位 TO W        (第一单位送 W)
  DO WHILE .NOT.EOF       (外循环判断文件是否
                            结束)
  STORE 0 TO S           (S 用于累加实发工资)
```

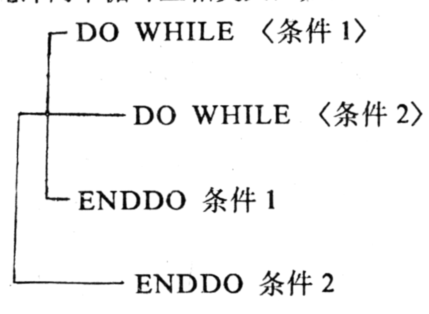
```
STORE W TO K        (K用于构成内循环条件)
DO WHILE K=单位         (相等表示是同一单位)
STORE S+实发工资 TO S      (累加实发工资)
DISPLAY                 (显示记录)
SKIP                    (下移一个记录)
IF K=单位               (比较前后两人是否同单位)
LOOP                    (相同则再转内循环)
ELSE                    (不相同则换单位)
STORE 单位 TO W         (将下一单位保存)
ENDIF
ENDDO
?"小计", S              (显示小计值)
ENDDO (∧W)
·DO  MCF13↓
```

输出数据略。这里主要说明两重循环的用法。对于这个具体问题，还有更为简便的解决方法，请读者自己考虑。

对循环语句来说，要注意以下几点：

①DO WHILE 和 ENDDO 必须一一对应。

②一个循环可以嵌套另一个循环，或两个循环并列。不允许两个循环互相交叉，如：



这是错误的循环结构。

③ 这里的循环同其它高级语言（如：BASIC, FORTRAN 等）的循环不完全相同。循环体内必须改变条件以期朝着循环结束的方向前进。

6.3 报表格式设计

在 cdBASEⅡ数据库中，实现报表格式设计有两个途径。一是利用 REPORT 命令由对话方式来建立；二是用…SAY…命令以程序的方式来实现。前者建立的是报表格式（.FRM）文件，后者建立的是格式（.FMT）文件。

6.3.1 报表格式文件

这种文件专用于提供输出数据的格式信息，是用对话方式来建立的。

1. 建立方式

格式: REPORT 〈文件名〉

作用: 允许用对话方式为活化数据库文件建立一个输出报表格式。

REPORT 命令可以建立两种报表格式：一是不带小计的报表；另一种是带有小计的报表。下面分别用例子来说明。

例 14. 对销售文件的数据，按如下的报表格式打印出来。

* * 销售情况统计表 * *

姓 名	品 名	单 价	数 量	金 额
张森林	电冰箱	1030.96	2	2061.92

……

操作过程是:

· *USE* 销售↓

· *REPORT*↓ (以下是对话内容)

输入格式报表名: 报表 *1*↓

输入选择, *M*=左页边, *L*=行数/页, *W*=页宽 *5,25,35*↓

页标题? (*Y/N*) *Y*↓

输入页标题: * * 销售情况统计表* *

双空格报表? (*Y/N*) *N*↓

需要总计吗? (*Y/N*) *N*

列　　　长度, 内容

001　　　*6*,姓名↓

输入报表标题: 姓名↓

002　　　*6*,品名↓

输入报表标题: 品名↓

003　　　*7*,单价↓

输入报表标题: 单价↓

004　　　*3*,数量↓

输入报表标题: 数量↓

005　　　*7*,金额↓

输入报表标题: 金额↓

006↓

各输入项的含义是:

输入格式报表名: ——要求用户回答报表文件的名字。本例中敲入: 报表 1。

输入选择, *M*=左页边, *L*=行数/页, *W*=页宽——

这是要求用户回答报表左边留多少个空格，每一页打印多少行，整个报表要占多少列（即页宽）。本例中输入：5,25,35。分别对应留空格 5 个，每页 25 行，页宽是 35 个字符。

页标题？（*Y / N*）——回答报表每页是否需要表头。本例回答：Y

输入页标题：——敲入页标题。本例是：* * 销售情况统计表 * *

双空格报表？（*Y / N*）——所谓双空格报表即打印一行空一行。询问是否需要打印一行空一行。本例回答：N，即不需双空格报表。

需要总计吗？*(Y / N)*——询问报表中是否需要将某些字段的值按列累加起来。本例回答：N（不需要）。

列长度内容——需要打印的数据库文件的字段。可以按照定义字段时的形式进行回答。第一个字段（用 001 表示，系统提示）的宽度是 6,"内容"位置下敲入"姓名"字段。本例敲入的是：6，姓名。

输入报表标题：——要求回答对应字段"姓名"的栏目是什么，即打印在纸上的栏目名字。这个名字可以和字段名相同，也可以不同，由用户自己确定。本例敲入：姓名。这表示字段名和格式栏目名称相同。我们也可以敲入"售货员姓名"作为报表的栏目。

其余类似。当出现一个新的字段号码提示（如：006），我们敲入↓，就表示报表格式定义完毕，系统将用户定义的报表格式信息保存在磁盘，然后回到点状态。这就建好了一个"报表 1"的报表格式文件。

需要强调两点：一是用户必须依次正确回答系统提问；

二是“长度，内容”位置只能回答相应的字段名，而“输入报表标题:”之后可以回答任何字符。

2. 报表格式文件的调用

格式：REPORT FORM 〈文件名〉〔TO PRINT〕

作用：调用报表格式文件，格式输出数据库内容。

例 15. 调用“报表 1”的格式来输出销售文件的数据。

· *USE* 销售↓

· *REPORT FORM* 报表 *1*↓

页号：*00001*

06 / 23 / 89

* * 销售情况统计表 * *

姓 名	品 名	单 价	数 量	金额
张森林	电冰箱	*1030.96*	*2*	*2061.92*
张森林	收录机	*715.74*	*1*	*715.74*
徐荣策	自行车	*179.93*	*5*	*899.65*
徐荣策	电视机	*1140.00*	*3*	*3420.00*
吕速才	电冰箱	*1030.96*	*2*	*2061.92*
吕速才	收录机	*715.74*	*3*	*2147.72*

3. 带小计的报表格式文件

如果我们希望数据库文件中的数据在按格式输出的同时，还能分组小计，那么可用带分类小计的报表格式文件。要设计这种格式，要求数据库文件必须按分类字段排序或索引。

例 15. 假设销售文件中的数据按例 14 的格式输出，且还要按品名来分类小计，即将同种商品的金额加起来。建立文件用到的语句如下：

· *USE* 销售↓

·*SORT ON 品名 TO SS*↓ (按品名排序)

·USE　SS↓

·REPORT↓

输入格式报表名：报表2↓

输入选择，M=左页边，L=行数/页，W=页宽 5,35,38↓

页标题？(Y/N) Y

输入页标题：销售小计表↓

双空格报表？(Y/N) N

需要总计吗？(Y/N) Y

报表里有小计吗？(Y/N) Y 输入需小计的字段：品名↓

只需简单报表？(Y/N) N

小计后换页？(Y/N) N

输入小计标题：商品名称：↓

列　　　长度，内容

001　　　　6，姓名↓

输入报表标题：姓名↓

002　　　　6，品名↓

输入报表标题：品名↓

003　　　　7，单价↓

输入报表标题：单价↓

需要总计吗？(Y/N) N

004　　　　3，数量↓

输入报表标题：数量↓

需要总计吗？(Y/N) N

005　　　　7，金额↓

输入报表标题：金额↓

需要总计吗？（Y／N） Y

006↓

这里有几项需要解释一下：

报表里有小计吗？（*Y／N*）——这个询问意思实际上是报表要按某个字段值来分组吗？回答 Y 表示要分组，N 表示不分组。本例回答：Y（要分组）

需要小计的字段：——紧接上问。意思是按哪一个字段来分组（小计）。我们可以按姓名来分组，即同一个人销售的商品打印在一起，并将其金额小计。也可以按品名分组，即将同种商品打印在一起，且将同种商品的销售金额累加起来。分组和排序（索引）的字段要相同。本例中敲入：品名(即按品名分类小计)。

只需简单报表？*(Y／N)*——所谓简单报表就是说报表中只包含分类小计项，其余细目项均不给出。简单报表又称摘要报表。本例回答：N（即分类项和细目项均要）。

小计后换页？（*Y／N*）——意思是每个小计项是否就作为报表的一页。本例回答：N

输入小标题：——对每一个分组给出标志。本例给的是：商品名称：

另外，在每个数值型字段之后均要求回答：

需要总计吗？（*Y／N*）——表示该字段是否要小计即按分组来小计。本例要求按金额小计，因而回答：Y

其余项目的含义与例 14 中相同。下面调用之。

· *USR SS*↓

· *REPORT FORM* 报表 *2*↓（以下是输出结果）

页号：*00001*

06/23/89

销售小计表

姓名	品名	单价	数量	金额
* 商品名称：电冰箱				
张森林	电冰箱	1030.96	2	2061.92
吕速才	电冰箱	1030.96	2	2061.92
* * 小计 * *				4123.84
* 商品名称：电视机				
徐荣策	电视机	1140.00	3	3420.00
* * 小计 * *				3420.00
* 商品名称：收录机				
张森林	收录机	715.74	1	715.84
吕速才	收录机	715.74	3	2147.72
* * 小计 * *				2863.46
* 商品名称：自行车				
徐荣策	自行车	179.93	5	899.65
* * 小计 * *				899.65
* * 总计 * *				11306.95

请读者结合输入问答和输出结果，弄清各回答项的含义。使用中注意：

①报表格式文件的修改同建立方式一样。只需将光标移到待修改的位置，重新敲入正确内容就行了。光标控制键请参见 MODIFY COMMAND 命令。

②带分类小计的报表格式文件对应的数据库文件必须在分类字段上是有序的。或者说，只有有序文件才能建立带分类小计的报表格式文件。

③报表格式文件建立的只能是一种简单的报表，它既不

能画横线，也不能画竖线。对嵌套栏目等复杂的表格也无能为力。

6.3.2 格式文件的建立和调用

从上节已经看到了，报表格式文件在实际应用中有很大的局限性。为了描述更为复杂的报表，cdBASEⅡ提供了更为有力的命令@…SAY…，它可用于建立更为复杂和实用的格式文件。

格式文件（.FMT）和报表格式文件是两种不同类型的文件。它们都可以用于建立输出数据的报表格式。格式文件中用到的主要命令是@。

格式： @〈行，列〉SAY〈表达式〉〔USING〈字型描述符〉

作用： 从给定的座标位置开始显示（或输出）表达式的值。

其中：〈行，列〉是屏幕或打印机的座标位置，它可以是数值常数或数值变量。〈表达式〉可以是字段名、内存变量名、常数、算术表达式、关系表达式、逻辑表达式以及字符串表达式。常用的是字段名和字符串常数。USING〈字型描述符〉用于规定表达式值的输出格式。

建立格式文件同命令文件一样，只是文件的扩展名不同而已。

例 16. 将商品文件的数据用格式显示出来。

```
·MODIFY  COMMAND  格式1,FMT↓
    @1, 0  SAY "                    "
    @2, 0  SAY "                    "
    @3, 0  SAY "                    "
```

```
@4, 0  SAY "|"
@4, 2  SAY 货号
@4, 8  SAY"|"
@4, 10 SAY 品名
@4, 16 SAY"|"
@4, 18 SAY 单位
@4, 26 SAY"|"
@4, 28 SAY 型号
@4, 34 SAY"|"
5, 1    SAY "
        (∧W)
 · USE  商品↓
 · SET FORMAT TO 格式 1.FMT↓
 · DISPLAY
```

货　号	品　名	单　价	型　号
115	自行车	179.93	环球-2

这种形式的格式文件用于屏幕显示比较方便和直观，它一反系统提供的标准格式。如果再结合 GET 和 READ，可构造比较满意的屏幕格式输入。这对实际应用来说是有意义的。

注意:

①格式文件中只允许两个命令，一个就是 …SAY …，另一个是注释 NOTE 或 REMARK 或 * 其格式和作用是:

格式：NOTE／REMARK／*〈字符序列〉

作用：对程序提供注解。〈字符序列〉就是注解内容。

例 17. NOTE 这是一个程序。

②SET FORMAT TO〈格式文件名.FMT〉表示打开已经建立的格式文件。

③这种@…命令最适宜于构造屏幕输入／出格式。

④MODIF COMMAND 建立格式文件时，文件名之后必须跟上扩展名.FMT。

6.3.3 实用报表设计

在实际应用中，往往是数据和格式交织在一起，因而更多的是用命令文件来刻划报表和装填相应的数据。下面举一个例子就清楚了。

例 18. 按以下格式打印工资文件。

姓　名	收　　入			支　　出		实发工资
	基本工资	福利费	奖金	房租费	水电费	
张森林	*73.46*	*2.53*	*7.00*	*1.89*	*2.31*	*78.79*
徐荣策	*68.35*	*3.10*	*7.15*	*1.45*	*1.55*	*75.95*
…	…	…	…	…	…	…

程序说明：

W——用于控制报表每页打印记录的个数。由于每个记录代表一个人的信息，因而也可以就是用于控制每页人数。

A——屏幕或打印机的行号。用于确定输出信息所在的行号。

P——生成报表的页码。一份报表可能有多页，每一页依次打印一个页码。

K——用于处理最后不满一页的情况，要求页号码也打印在每页的右下角。

L——统计每页人数是否为 W。如果等于 W，则打印页码，否则继续循环，这意味着本页还不满 W 个记录。

·MODIFY COMMAND MCF18↓

```
NOTE 这是报表的例子。
ERASE                    (请屏幕)
INPUT"输入每页记录数:"TO  W
STORE  W  TO  K
A=1
P=0
USE  工资
DO WHILE .NOT.EOF
L=1
P=P+1
@A, 20 SAY "工资情况一览表"
A=A+1
@A, 59 SAY "单位: 元"
A=A+1
DO  LL
A=A+1
@A, 0 SAY "  "
@A, 10 SAY"      收      入"
```

```
@A, 40 SAY "|       支     出"
@A, 60 SAY"|         |"
A=A+1
@A, 0 SAY "|  姓名"
@A, 10 SAY "|——————————————"
@A, 40 SAY "|——————————————"
@A, 60 SAY "|  实发工资   |"
A=A+!
@A, 0 SAY "|                "
@A, 10 SAY "|  基本工资"
@A, 21 SAY "|  福利费"
@A, 31 SAY "|  奖金"
@A, 40 SAY "|  房租费"
@A, 50 SAY "|  水电费"
@A, 60 SAY "|           |"
A=A+1
DO  LL
A=A+1
DO  WHILE  L<=W
@A, 0 SAY "|"
@A, 2 SAY 姓名
@A, 10 SAY"| "
@A, 12 SAY 基本工资
@A, 21 SAY " | "
@A, 22 SAY 福利费
@A, 30 SAY " | "
@A, 31 SAY 奖金
```

```
@A, 40 SAY "|"
@A, 41 SAY 房租费
@A, 50 SAY "|"
@A, 51 SAY 水电费
@A, 60 SAY "|"
@A, 62 SAY 实发工资
@A, 70 SAY "|"
A=A+1
DO  LL
L=L+1
A=A+1
SKIP
IF  EOF
STORE  L  TO  K
STORE  W  TO  L
ENDIF
ENDDO
IF  K<W
A=A+W-K
ENDIF
A=A+2
@ A, 50 SAY "第"
@ A, 52 SAY P
@ A, 56 SAY "页"
A=A+4
ENDDO       (∧W)
·MODIFY CIMMAND LL↓
```

```
NOT  打印一条横线
@ A O SAY "————————"
@ A, 24 SAY "————————"
@ A, 45 SAY "————————"
RETURN      (∧W)
·SET FORMAT TO PRINT↓ (连通打印机)
·DO MCF18↓
```

输入每页记录数: 3↓

工资情况一览表

单位: 元

姓　名	收　　入			支　　出		实发工资
	基本工资	福利费	奖金	房租费	水电费	
张森林	73.46	2.53	7.00	1.89	2.31	78.97
徐荣策	68.35	3.10	7.50	1.45	1.55	75.95
吕速才	68.92	3.00	8.80	2.11	1.98	76.63

第 1 页

工资情况一览表

单位: 元

姓　名	收　　入			支　　出		实发工资
	基本工资	福利费	奖金	房租费	水电费 C	
王锡刚	88.87	3.56	8.88	1.48	2.60	97.23
陈力	125.67	3.55	8.66	1.99	2.88	133.01

第 2 页

由此例可以看出，不管报表格式有多么复杂，只要用@〈行，列〉SAY 精确地描述出其行列应该对应的位置就行了。用@命令还可以设计屏幕显示菜单。

例 19. 假设一个大型程序的总控程序流程如下：

总控程序

1	2	3	4	5	0
输入数据	修改数据	更新文件	统计数据	打印报表	退出运行

用菜单形式来提示总控程序的各种操作。这里假定“输入数据”对应 PL1 子程序，“修改数据”对应 PL2，其余操作类推。程序如下：

```
· MODIFY COMMAND MCF19↓
  ERASE
@1, 2 SAY "┌──────────────────────────────┐"
@2, 2 SAY "│   1.输入数据      2.修改数据   │"
@3, 2 SAY "│   3.更新文件      4.统计数据   │"
@4, 2 SAY "│   5.打印报表      0.退出运行   │"
@5, 2 SAY "└──────────────────────────────┘"
@6, 2 SAY "请选择代码:"
WAIT TO DM
DO CASE
```

```
DO DM="1"
DO PL1
CASE DM="2"
DO PL2
CASE DM="3"
DO PL3
CASE DM="4"
DO PL4
CASE DM="5"
DO PL5
CASE DM="0"
QUIT
ENDCASE
RETURN        (∧W)
```

当然，子程序中还可以用菜单，这样一层一层的提示用户，使得不懂计算机的人，也容易使用程序。

在设计实用报表时注意：

①尽量美观、大方、实用、上下竖线、栏目对齐。

②在@命令中屏幕上行列必须满足 0<行号<9，0<列号<34。在打印机上，行列可以是：0～254，但各种型号的打印机可能略有不同。

③在打印机上输出时，@命令中后面的行号应大于前面的行号，因为目前使用的打印机是一种"永向前"式打印机，打印头只能从上到下，从左到右依次打印。

④在打印机上输出报表之前，事先一定要打开打印机电源，且在敲 DO 命令之前先敲入：

· SET FORMAT TO PRINT ↓

习题六

1. cdBASEⅡ的对话和程序设计两种工作方式之间有哪些区别？

2. 举一个能够用到多重调用的列子，并写出完整的程序。

3. 用于定义内存变量的命令至少有 7 个，请一一列出。

4. WAIT, ACCEPT 和 INPUT 三者之间有什么区别?

5. …SAY…能否用于定义内存变量？为什么？

6. 举一个最适宜于 CASE 命令的例子，写出完整的语句，再将 CASE 改用 IF 语句来实现同样功能。

7. DO WHILE 和 DO〈命令文件〉各有什么不同?

8. 如果要将数据库文件中的数值型数据累加起来，至少有四种不同的办法来实现，请举一例子，然后将四种方法分别写出来。

9. 自己给出一个带有数据的表格，再用两种不同的方法来生成报表。

10. 举一个屏幕使用格式文件的例子，写出格式文件并调用之。

11. 举一个至少有三层嵌套的报表的例子，用命令文件的形式写出报表格式。

12. 假设数据库文件结构如下:

销售（姓名，品名，单价，数量、金额）

商品（品名，型号，库存量，产地）

编写以下问题的应用程序:

①找出销量在1000件且销售的合计金额超过20万元的商品名称及其产地。

②找出销售了张三所售的某种商品的其他售货员的名字。

③找出所有滞销的商品（销量在10件以下的商品都叫滞销商品)。

13. 有哪些命令可以用来生成数据库？请写出这些命令并说明哪种情况下适用。

14. 哪些命令的操作必须要涉及到两个工作区？请用例子说明它们各自的用法。

第七章　cdBASEⅡ数据库应用实例

本章采用由小到大，由简到繁的原则，介绍几个数据库应用系统。使读者逐步掌握应用汉字 dBASE 数据库系统来解决实际问题的能力。

对于初学者来说，在学习掌握前面各章内容的基础上，还需注意多看一些实例，摸索编程的规律和技巧。本章介绍的各个程序都是在一些单位投入运行的实用软件。

7.1 快速盘点系统

本程序按用户经营商品的单价分别接收并储存当天的增(进货)、减（销售）数据，自动快速计算输出当日的结存数量与金额，让经营者及时掌握盈亏情况，提高经营水平和竞争能力。此外，程序还可随时提供增减商品表，未增减商品表，供经营者分析决策。

7.1.1 快速盘点数据库文件（KSPD.DBF）结构

数据库结构　：A：KSPD.dbf

记录个数　　：6

最后更新日期：01／01／88

字段	字段名	类型	宽度	小数
1	单价	Numeric	8	4
2	昨日结存	Numeric	5	
3	本日增加	Numeric	5	
4	本日减少	Numeric	4	
5	本日结存	Numeric	5	
6	结存金额	Numeric	10	4
* *总计* *			38	

7.1.2 程序功能结构图

快速盘点程序功能结构如图 7—1 所示

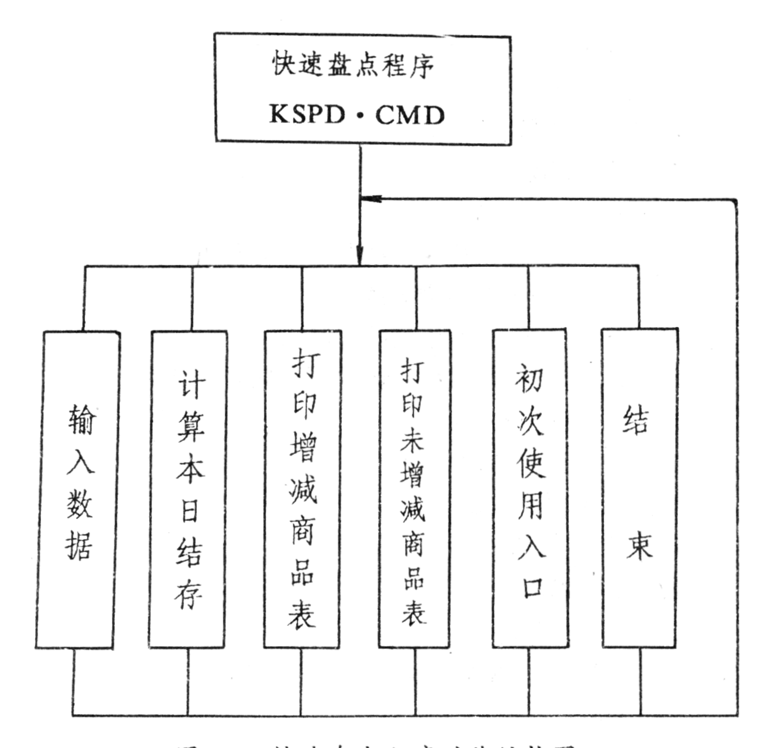


图 7—1 快速盘点程序功能结构图

7.1.3 源程序清单及解释

1 ＊这是一个快速盘点结算程序

2 STOR 'N' TO HD

3 DO WHILE HD='N'

4 ERASE

5 ACCEPT "请输入日期（××·××·××）"
TO RQ

```
6  ?  "日期: &RQ 正确否 (Y/N)"
7  WAIT   TO   HD
8  ENDDO
9  DO   WHILE   1<2
10  ERASE
11  ?  "1……输入数据"
12  ?  "2……计算本日结存数量与金额"
13  ?  "3……打印增减商品表"
14  ?  "4……打印未增减商品表"
15  ?  "5……初次输入数据 (要输昨日结存)"
16  ?  "0……结束"
17  ?  "请选择!"
18  WAIT    TO    XZ
19  USE    KSPD
20  ERASE
21  DO   CASE
22  CASE    XZ='0'
23  USE
24  CANCEL
25  CASE    XZ='1'.OR.XZ='5'
26  IF   XZ='5'
27  DELETE    ALL
28  PACK
29  ENDIF
30  IF   XZ='1'
31  REPL   ALL  昨日结存 WITH 本日结存,
      本日增加 WITH 0, 本日减少 WITH 0
```

```
32  ENDIF
33  COUNT  TO  JL1
34  STORE  'Y'  TO  HD
35  DO  WHILE  HD='Y'
36  STORE  'N'  TO  XD
37  STORE  0  TO  JC
38  STORE  0  TO  DJ
39  STORE  0  TO  ZJ
40  STORE  0  TO  JS
41  DO  WHILE  XD='N'
42  ERASE
43  IF  XZ='5'
44  @ 2, 20 SAY "请输入昨日结存" GET JC
45  ENDIF
46  @ 3, 20 SAY "请输入单价" GET DJ
47  @ 4, 20 SAY "请输入本日增加" GET ZJ
48  @ 5, 20 SAY "请输入本日减少" GET JS
49  READ
50  @ 6, 20  SAY "请校对，正确否 (Y/N)"
      GET  XD
51  READ
52  ENDDO
53  LOCATE  FOR  单价=DJ
54  IF  EOF
55  APPEND  BLANK
56  REPL 单价 WITH DJ, 昨日结存 WITH JC
57  ENDIF
```

```
58  REPL 本日增加 WITH ZJ,
         本日减少 WIHT JS
59  ERASE
60  @7, 20  SAY "继续输入吗? (Y/N)"
    GET  HD
61  READ
62  ENDDO
63  COUNT  TO  JL2
64  IF  JL1<JL2
65  INDEX  ON  单价  TO  KSPDIN
66  ENDIF
67  CASE  XZ='2'
68  USE  KSPD
69  REPL ALL 本日结存 WITH 昨日结存+
         本日增加-本日减少
70  REPL  ALL  结存金额 WITH 单价*本日结存
71  CASE  XZ='3'
72  USE  KSPD  INDEX  KSPDIN
73  STORE  "本日增加<>0.OR.本日减少
    <>0"  TO  TJ
74  SET  PRINT  ON
75  ?  "  "
76  ?  "  "
77  ?  "          &RQ 增减商品表"
78  LIST OFF 单价, 本日增加, 本日减少, 本日结
    存, 结存金额 FOR &TJ
79  ?  "  "
```

```
80  SET  PRINT  OFF
81  SUM  结存金额  TO  JE  FOR  &TJ
82  STORE  0  TO  ZJ
83  STORE  0  TO  JS
84  GO  TOP
85  DO  WHILE .NOT.EOF
86  STORE  单价*本日增加+ZJ  TO  ZJ
87  STORE  单价*本日减少+JS  TO  JS
88  SKIP
89  ENDDO
90  SET  PRINT  ON
91  ?  “  ”
92  ?  “本日增加总额”，ZJ,“减少总额”，JS
93  ?  “增减结存总额”，JE
94  ?  “  ”
95  SET  PRINT  OFF
96  CASE  XZ=‘4’
97  USE  KSPD  INDEX  KSPDIN
98  STORE  “本日增加=0.AND.本日减少=0”
    TO  TJ
99  SET  PRINT  ON
100  ?  “  ”
101  ?  “  ”
102  ?  “            &RQ 未增减商品表”
103  LIST 单价，本日结存，结存金额
         OFF FOR &TJ.AND.本日结存<> 0
104  ?  “  ”
```

```
105  SET  PRINT  OFF
106  SUM  结存金额  TO  WJE  FOR  &TJ
107  SET  PRINT  ON
108  ?  "未增减结存总额", WJE, "总计结存额",
     JE+WJE
109  ?  "  "
110  SET  PRINT  OFF
111  ENDCASE
112  ENDDO
113  RETURN
```

注意：各语句前的序号是为了解释加注的，运行的程序中是不应该有的。

本程序第1句是一个注释语句，注明本程序是一个快速盘点结算程序。第2～8句提示并接受用户输入日期，采用提示用户校对输入日期的办法，保证输入日期的正确性。循环结构直到用户确认输入的日期正确，按Y键中止循环。宏替换&RQ将变量RQ接受的日期替换在语句中的&RQ处，如：输入为：89.05.11↓，则执行第6句的屏幕显示为:

日期：89.05.11 正确否（Y/N）

若用户认为正确按Y键；否则按N键，再重输日期。

第9～112句是程序主体，为让用户自由随意选择各项功能，除"0"以外的1～5各项功能选择运行后，均重现功能显示，供用户选择执行。故使用循环语句，只有当用户选择"0"，才结束循环并使整个程序的运行停止。第10句和前面的第4句是清屏幕的语句，作用是清除执行本语句前屏幕上的显示。第11～16句是本程序的功能菜单。17句提示用户

选择 11～16 句中的功能。19 句是打开数据库 KSPD.DBF。21 句是区别各种功能情形的情形语句头。22 句判断选择，若为"0"则执行 23 句关闭数据库，24 句终止程序运行，回到数据库系统点（·）提示符。

第 25 句是选择功能 1 或 5 的情形入口。26～29 句完成初次使用的初置工作，即删去旧（或称表演）数据。30～32 句，对通常输入数据时，首先将前一天的本日结存列入昨日结存，并将前一天的增加减少清零（以免昨天的增减数引起混乱）。33 句统计并保存前一天的记录数，以备后用。34 句为使 35 句的首次循环条件为真而给变量 HD 置初值。第 36 句同 34 句，37～40 句为工作变量清零（0），41 句～52 句为校对修改输入错误的循环，其方法与 3～8 句对输入日期的处理类似，其中 43～45 句供初次使用输入昨日结存。

第 53 句寻找库中单价与输入单价相等的记录。54 句判断是否文件尾，如是文件尾，说明库中无本次输入单价的记录，则由 55 句添加一个空记录，将输入数据通过 56、58 句装入新记录；如不是文件尾，说明输入单价在数据库中已经存在，则由 58 句装入本日增加和减少。由执行 60 句时用户的回答（Y／N）决定 36～62 句是否重复执行。

第 63 句统计并保存输入数据之后的记录数。64 句比较输入数据前后的记录数，判断是否增加了记录，如输入后的记录数 JL2 大于 JL1，则执行 65 句索引数据库。便于选择 3 或 4 时按索引库打印商品表。

第 71 句是选择 2（计算结存）的入口，69 和 70 句完成结存数量与金额的计算。

第 71 句是选择 3（打印增减商品表）的入口，72 句打开索引库。73 句赋字符串值到变量 TJ，供 78 句和 81 句作

条件使用。74句连通打印机，80句断开打印机，下同。对于未经扩充，只有一个扩充槽插入Z_{80}卡使用数据库的情况下，对这种与打印机有关的语句前加一个星号“*”作为注释句。75句和76句使打印或显示空走两行。77句打印表头题目。78句打印列出本日有增减变化的记录（因每天输入前增、减通通置为零，输入后打印或显示时不等于零，就表示有增加或减少）。81句合计增、减结存额，82～89合计增加、减少总额，92～93句打印这三个值。

第96句是选择4（打印未增减表）的入口，98句将103句和106句中用的条件送入变量TJ，103句按单价列出未增减商品的结存数量与金额。106句合计未增减商品结存额，108句打印未增减商品结存及增减和不增减商品总金额。从108句总金额由JE+WJE可见，必须先打印商品增减表（计算JE），才能再打印未增减表中的总结存额。

7.1.4 说明:

本程序比较简短，比较适宜初学者学习。本程序由成都三开元电脑部组织开发并投入使用，为满足经营决策及业务需要，历经本部不断的扩充完善，已发展成为商品经营的综合管理软件，实现了从商品入库、调拨、库存、销售（批、零)、汇总等全面微机管理。

7.2 住房管理系统

本程序是应某部房管处需要开发的。适用于一般单位的职工住房数据信息的输入、修改、分类、检索及统计。可为用户在收房租、计划新建、改建和分配职工住房等工作中提

供决策参考信息。

程序简单明了，适应初学者，易于维护使用。

7.2.1 住房管理数据库（ZFGLK.DBF）结构

数据库结构　：a：zfglk.dbf

记录个数　　：4

最后更新日期：05／16／88

字段	字段名	类型	宽度	小数
1	住址	Character	10	
2	建面积	Numeric	5	1
3	用面积	Numeric	5	1
4	租金	Numeric	6	2
5	间	Character	2	
6	费	Character	2	
7	部别	Character	8	
8	姓名	Character	6	
9	职务	Character	8	
10	条件	Character	2	
11	进时	Character	6	
总计			61	

其中：字段住址的值为房屋的房号，由幢、单元、楼层、门号之代码构成，共计10个字符；建面积即住房的建筑面积；用面积即住房的使用面积；费为收费方式，其值是：扣／收／非（即记录中字段费的内容是扣、收、非三者之一，扣即从工资中扣收，收即收款，非即非扣非收的其它方式）。字段条件为职工居住房屋的条件，其值可以是在、余、离、退、调、烈、转之一。字段进时为住户住进房屋的

时间（年月）。

7.2.2 住房管理软件功能结构图

住房管理软件功能结构如图 7–2 所示。

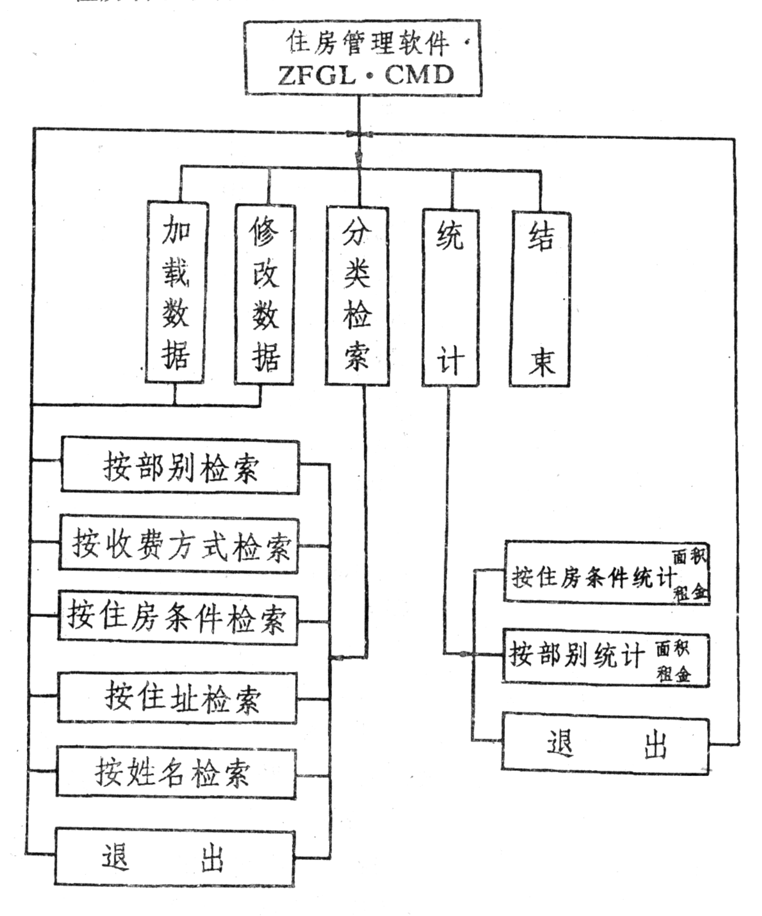


图 7–2 住房管理软件功能结构图

7.2.3 源程序清单

```
1   * 这是一个住房管理程序
2   DO  WHILE  1=1
3   ERASE
4   ?  "1……录入"
5   ?  "2……修改"
6   ?  "3……检索"
7   ?  "4……统计"
8   ?  "0……结束"
9   ?  "请选择"
10  WAIT  TO  XZ
11  STORE  "Y"  TO  HD
12  USE  ZFGLK
13  DO  CASE
14  CASE  XZ='0'
15  CANCEL
16  CASE  XZ='1'
17  DO  WHIL  HD='Y'
18  APPEND  BLANK
19  ?  '记录号', #
20  EDIT
21  ?  '继续吗 (Y/N)'
22  WAIT  TO  HD
23  IF  HD='N'
```

```
24  USE
25  ENDIF
26  ENDDO
27  CASE  XZ='2'
28  DO  WHILE  HD='y'
29  ERASE
30  ACCPT  '请输入修改的房号'  TO  FH
31  LOCATE  FOR  住址=FH
32  ?  '记录号', #
33  EDIT
34  ?  '继续修改吗 (Y/N)'
35  WAIT  TO  HD
36  IF  HD='N'
37  USE
38  STORE  'N'  TO  HD
39  ENDIF
40  ENDDO
41  CASE  XZ='3'
42  STORE  '1'  TO  HD
43  DO  WHILE  HD='1'
44  ERASE
45  ?  '1……部别'
46  ?  '2……收费方式'
47  ?  '3……住房条件'
48  ?  '4……住址'
```

```
49  ?  '5……姓名'
50  ?  '0……退出'
51  ?  '请选择检索方式'
52  WAIT  TO  XZ2
53  ERASE
54  ?  '检索结果是打印/显示 (Y/N)'
55  WAIT  TO  DY
56  IF  DY='y'
57  SET  PRINT  ON
58  ELSE
59  SET  PRINT  OFF
60  ENDIF
61  GO  1
62  DO  CASE
63  CASE  XZ2='1'
64  ACCEPT  '输入部别名'  TO  BB
65  DO  WHILE  .NOT.EOF
66  LIST  NEXT  10  OFF  FOR
    @('&BB', 部别) >0
67  ?  '按任一键继续'
68  WAIT
69  ENDDO
70  CASE  XZ2='4'
71  ACCEPT  '输入住址'  TO  ZZ
```

```
72  GO  1
73  DO  WHILE  .NOT.EOF
74  LIST  NEXT  OFF  10  FOR  @('&ZZ',
    住址) >0
75  ?  '按任一键继续'
76  WAIT
77  ENDDO
78  CASE  XZ2='5'
79  ACCEPT  '请输入姓名'  TO  XM
80  LIST  OFF  FOR  @('&XM', 姓名) >0
81  ?  '按任一键继续'
82  WAIT
83  CASE  XZ2='2'
84  ACCEPT  '请输入收费方式 (扣/收/非)'
    TO  SF
85  DO  WHILE  .NOT.EOF
86  LIST  NEXT  10  OFF  FOR  @
    ('&SF', 费) >0
87  ?  '按任一键继续'
88  WAIT
89  ENDDO
90  CASE  XZ2='3'
91  ACCEPT  '输入住房条件 (在/余/离/退/
    调/烈/转)'  TO  TJ
```

```
92  DO  WHILE  .NOT.EOF
93  LIST  NEXT  10  OFF  FOR
    @ (&TJ', 条件) >0
94  ? '按任一键继续'
95  WAIT
96  ENDDO
97  CASE  XZ2='0'
98  STORE  '0'  TO  HD
99  ENDCASE
100  IF  DY='y'
101  SET  PRINT  OFF
102  ENDIF
103  ENDDO
104  CASE  XZ='4'
105  STORE  '1'  TO  HD
106  DO  WHILE  HD='1'
107  ERASE
108  ?  '1……按住房条件统计面积与租金'
109  ?  '2……按部别统计面积与租金'
110  ?  '0……退出'
111  ?  '请选择!'
112  WAIT  TO  XZ3
113  DO  CASE
114  CASE  XZ3='1'
```

```
115 ACCEPT '请输入住房条件（在/余/离/
    退/调/烈/转)' TO TJ
116 SUM 建面积，用面积，租金 TO  JM，SM，
    ZJ  FOR  @('&TJ'，条件) >0
117 ? '建筑面积       使用面积        租金'
118 ? JM，SM，ZJ
119 ? ' '
120 CASE  XZ3='2'
121 ACCEPT '请输入部别'  TO  BB
122 SUM 建面积，用面积，租金 TO  JM，SM，
    ZJ  FOR  @('&BB'，部别) >0
123 ? '住房部别为：&BB'
124 ? '建筑面积   使用面积     租金'
125 ? JM，SM，ZJ
126 ? ' '
127 CASE  XZ3='0'
128 STORE  '0'  TO  HD
129 ENDCASE
130 IF  Dy='y'
131 SET  PRINT  OFF
132 ENDIF
133 ENDDO
134 ENDCASE
135 ENDDO
```

136 RETURN

本程序比较直观简明，不必逐句解释，于此仅对可能费解的地方解释如下:

1.第 19 句的作用是，当执行本句时，屏幕显示当前记录号，因为 20 句 EDIT 命令执行时，要求提供记录号，用户照执行 19 句时显示的记录号键入后，则按提示一个字段一个字段的输入到添加的空记录中。

2.程序的检索部分，（情形）CASE 语句没有依菜单出现的顺序出现 1～5 种情形，在 63 句是 CASE XZ2＝‘1’，70 句是 CASE XZ2＝‘4’，83 句才是 CASE XZ2＝‘2’，这是允许的，不影响执行效果。

3.在统计程序段中的 117 句是为在 118 句显示统计数字的上面，显示对应的名称，这些名称的位置根据试运行时 118 句统计数字的显示位置，用插入空格来调整。

4.第 65 句至 69 句的程序功能是每屏显示 10 个符合条件的记录，用户看清后任按一键，又循环显示下一屏符合条件的记录，如此直到文件结束。

7.3 工资管理软件

考虑到中华学习机的广泛普及。一方面为满足读者学习需要；一方面为了让用户更好地发挥该机作用。将这个工资管理软件作为本节一个实例。为照顾初学者简明易读。省去了一般工资软件不具备的汇总和劳资统计部分。只需将本软件的库结构及有关的几个语句作相应改动，即可适用于要求和习惯不同的单位。本软件经地方和军队多个用户使用实际说明其实用性、通用性好。

7.3.1 工资管理软件数据库（GZK.DBF）结构

数据库结构　：B: GZK.dbf

记录个数　　：33

最后更新日期：01／01／88

字段	字段名	类型	宽度	小数
1	号	Character	3	
2	姓名	Character	8	
3	基本工资	Numeric	9	2
4	工龄工资	Numeric	9	2
5	粮贴	Numeric	7	2
6	民族贴	Numeric	6	2
7	驾驶	Numeric	6	2
8	工资合计	Numeric	10	2
9	四种补贴	Numeric	8	2
10	独子费	Numeric	7	2
11	自行车	Numeric	6	2
12	其它 1	Numeric	8	2
13	合计	Numeric	10	2
14	储蓄	Numeric	8	2
15	储金	Numeric	8	2
16	其它 2	Numeric	7	2
17	国库卷	Numeric	8	2
18	扣计	Numeric	9	2
19	实发数	Numeric	10	2
20	签章	Character	1	
＊＊总计＊＊＊			149	

7.3.2 工资管理软件功能结构图

工资管理软件功能结构如图 7–3 所示。

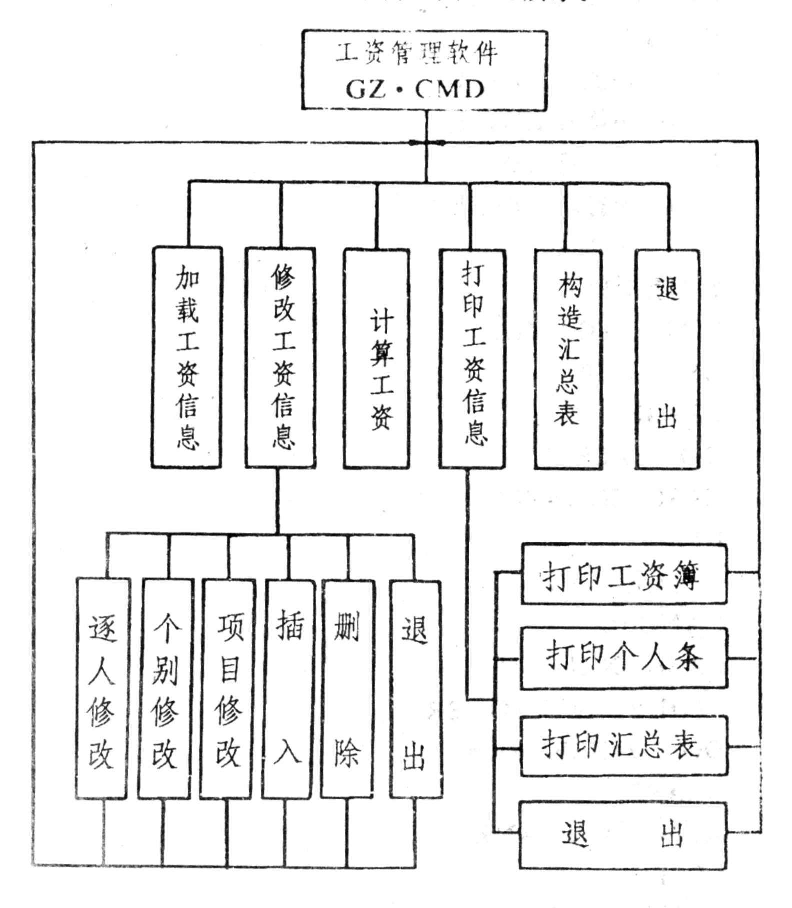


图 7–3 工资管理软件功能结构图

7.3.3 源程序清单

```
*这是一个工资处理程序
DO  WHILE  1<2
ERASE
? '1……加载工资信息'
? '2……修改工资信息'
? '3……计算工资'
? '4……打印工资信息'
? '0……结束'
? '请选择!'
WAIT  TO  XZ1
USE  GZK
ERASE
DO  CASE
CASE  XZ1='0'
USE
? '再见'
CANCEL
CASE  XZ1='1'
STORE  'y'  TO  SR
DO  WHILE  SR='y'
ERASE
APPEND  BLANK
EDIT
? "继续输入吗 (Y/N)"
WAIT  TO  SR
ENDDO
```

```
CASE  XZ1='2'
GO  TOP
STORE  'y'  TO  XG
DO  WHILE  XG='y'
ERASE
? '1……逐人修改'
? '2……个别修改'
? '3……项目修改'
? '4……插入某人工资信息'
? '5……删除某人工资信息'
? '6……使记录号=工号'
? '0……退出'
? '请选择!'
WAIT  TO  XZ2
ERASE
DO  CASE
CASE  XZ2='0'
USE
STORE  'N'  TO  XG
LOOP
CASE  XZ2='1'
GO  TOP
EDIT
CASE  XZ2='2'
INPUT  "请输入修改人员工号"
  TO  GH
GO  GH
```

```
EDIT
CASE  XZ2=‘3’
GO  TOP
STORE  “Y”  TO  XG
DO  WHILE  XG=‘y’
ERASE
? “1……按项目逐人修改”
? “2……按项目个别修改”
? “3……按项目同值全员修改”
? “4……每个人工龄工资增加 0.5 元”
? “0……退出”
? “请选择!”
WAIT  TO  XZ3
ERASE
IF  XZ3<>‘0’
ACCEPT  “请输入修改项目名:”
 TO  ZDM
ENDIF
DO  CASE
CASE  XZ3=‘0’
STORE  ‘N’  TO  XG
LOOP
CASE  XZ3=‘1’
GO. TOP
CHANGE  FIELDS 号, 姓名, &ZDM
CASE  XZ3=‘2’
STORE  ‘y’  TO  HD
```

```
DO  WHILE  HD='y'
GO  TOP
INPUT  "请输入修改人工号"
  TO  GH
GO  GH
INPUT  "请输入新数据"  TO  XSJ
REPL  NEXT  1  &ZDM  WITH  XSJ
? "继续修改吗 (Y/N)"
WAIT  TO  HD
ENDDO
CASE  XZ3='3'
INPUT  "请输入新内容"  TO  HM
REPL  ALL  &ZDM  WITH  HM
CASE  XZ3='4'
REPL  ALL  工龄工资  WITH  工龄工资
      +0.5  FOR  工龄工资<20
ENDCASE
ENDDO
CASE  XZ2='4'
INPUT  "请输入要插入位置之职工的工号"
TO  GH
GO  GH
INSERT  BEFORE
CASE  XZ2='5'
INPUT  "请输入要删除职工的工号"  TO  GH
GO  GH
DELETE  NEXT  1
```

```
PACK
CASE  XZ2='6'
GO  TOP
DO  WHILE .NOT.EOF
REPL  号  WITH  STR (#, 3)
SKIP
ENDDO
ENDCASE
ENDDO
CASE  XZ1='3'
DO  GZJS
CASE  XZ1='4'
STORE  'y'  TO  HD
DO  WHILE  HD='y'
ERASE
? "1…打印工资簿"
? "2…打印发给个人的工资单"
? "0…退出"
? "请选择!"
WAIT  TO  XZ4
USE  GZK
ERASE
DO  CASE
CASE  XZ4='0'
STORE  "N"  TO  HD
CASE  XZ4='1'
STORE  'y'  TO  JX
```

```
DO  WHILE  JX='y'
GO  TOP
INPUT  "请输入本次打印之第一人的工号"
 TO  GH1
STORE  GH1  TO  JLH1
INPUT  "请输入本次打印之最后一人的工号"
TO  GH2
STORE  GH2  TO  JLH2
N=JLH2-JLH1+1
GO  JLH1
DO  GZDY
?"继续打印吗? (Y/N)"
WAIT  TO  JX
ENDDO
CASE  XZ4='2'
USE  GZK
INPUT  "请输入打印第一人的工号"  TO  GH1
INPUT  "请输入打印人数"  TO  N1
STORE  GH1  TO  JLH1
GO  JLH1
SET  DEVICE  TO  PRINT
?"请调好纸后，任敲一键"
WAIT
SET  PRINT  ON
DO  WHILE  N1>0
SET  PRINT  ON
LIST  NEXT 1  OFF  TO  PRINT
```

```
?"      "
SET  PRINT  OFF
SKIP
N1=N1-1
ENDDO
?"      "
ENDCASE
ENDDO
ENDCASE
ENDDO
RETURN
```

计算子程序清单如下：

```
* 这是工资计算子程序，名字：GZJS.CMD
REPL  ALL 工资合计 WITH 基本工资+工龄工资
  +粮贴+民族贴+驾驶
REPL  ALL 小计 WITH 工资合计+四种补贴+独子费
  +自行车+其它1
REPL  ALL 扣计 WITH 储蓄+储金+其它2+国库卷
REPL  ALL 实发数 WITH 小计-扣计
RETURN
```

打印工资表子程序清单如下：

```
* 这是打印工资表子程序，名字：GZDY.CMD
STORE  N  TO  N1
SET  DEVICE  TO  PRINT
STORE  0  TO  HS
DO  WHILE  N>0
STORE  HS+1  TO  HS
```

```
IF  VAL (XZ4) <2
IF  HS=1
SET  PRINT  OFF
SET  CONSOLE  ON
?“请调纸到新页开始”
WAIT
REASE
SET  CONSOLE  OFF
@1, 1 SAY  CHR (27) +‘IB’
@2, 20 SAY“         工资发放表”
SET  PRINT  OFF
@3, 1 SAY  CHR (27) +‘IA’、
@4, 9 SAY “          年  月  日”
@4, 139 SAY “第       页共      页”
@5, 1  SAY  “==……==”
@5, 56  SAY  “==……==”
@5, 111  SAY  “==……==”
@5, 161  SAY  “=====”
SET  PRINT  ON
?“号  姓  名  基本工资  工龄工资  粮;
贴  民族贴  驾驶 工资合计 四种补贴;
独子费  自行车  其它1   小计   储蓄;
储金  其它2  国库卷  扣计  实发数  签章”
?“……………”
LIST  OFF  NEXT  1  TO  PRINT
STORE  9  TO  I
@ I, 1  SAY  “………”
```

```
ELSE
LIST  OFF  NEXT  1  TO  PRINT
STORE  I+1  TO  I
@ I, 1  SAY  "........."
ENDIF
ENDIF
SET  PRINT  OFF
IF  HS=22.OR.N=1
DO  GZHJDY
STORE  N-1  TO  N
SKIP
ENDDO
SET  CONSOLE  ON
SET  PRINT  OFF
RETURN
```

构造并打印合计子程序清单如下：

```
* 本子程序构造并打印合计，程序名：GZHJDY
DO  CASE
CASE  N=1
STORE  HS  TO  N1
CASE  HS=22
STORE  22  TO  N1
ENDCASE
GO  JLH1
SUM  NEXT  N1 基本工资，工龄工资，粮贴，民;
族贴，驾驶，工资合计，国库卷 TO J1, J2, J3, J4,
J5,
```

```
J6, J21
GO  JLH1
SUM NEXT N1  四种补贴，独子费，自行车，其;
它1，小计，储蓄，储金，其它2，扣计，
实发数 TO J8, J9, J10, J11, J12, J13, J14, J15, J16, J17
IF  HS=22
STORE  JLH1+22  TO  JLH1
ENDIF
STORE  I+1  TO  I
SET  PRINT  ON
?"本页合计"
@ I, 14  SAY  STR (J1, 9, 2)
@ I, 23  SAY  STR (J2, 9, 2)
@ I, 32  SAY  STR (J3, 7, 2)
@ I, 39  SAY  STR (J4, 6, 2)
@ I, 45  SAY  STR (J5, 6, 2)

@ I, 51  SAY  STR (J6, 10, 2)
@ I, 61  SAY  STR (J8, 8, 2)
@ I, 69  SAY  STR (J9, 7, 2)
@ I, 76  SAY  STR (J10, 8, 2)
@ I, 84  SAY  STR (J11, 8, 2)
@ I, 92  SAY  STR (J12, 10, 2)
@ I, 102  SAY  STR (J13, 9, 2)
@ I, 111  SAY  STR (J14, 8, 2)
@ I, 119  SAY  STR (J15, 8, 2)
@ I, 127  SAY  STR (J21, 8, 2)
```

```
@ I, 135  SAY  STR (J16, 9, 2)
@ I, 144  SAY  STR (J17, 10, 2)
IF  VAL (XZ4) <>3
@ I+1, 1  SAY  "==……=="
SET  PRINT  ON
? "       "
SET  PRINT  OFF
ENDIF
STORE  0  TO  HS
ENDIF
RETURN
```

7.4 汉英信息检索系统

7.4.1 系统功能

本软件采用模块化设计技术，按各种不同的逻辑功能，使用不同的模块实现。

系统提供 8 种建库及维护功能，5 种检索功能，可归纳如下：

1.建库功能。实现计算机联机信息检索要建立庞大的信息和检索库，本系统提供了方便的建库功能。

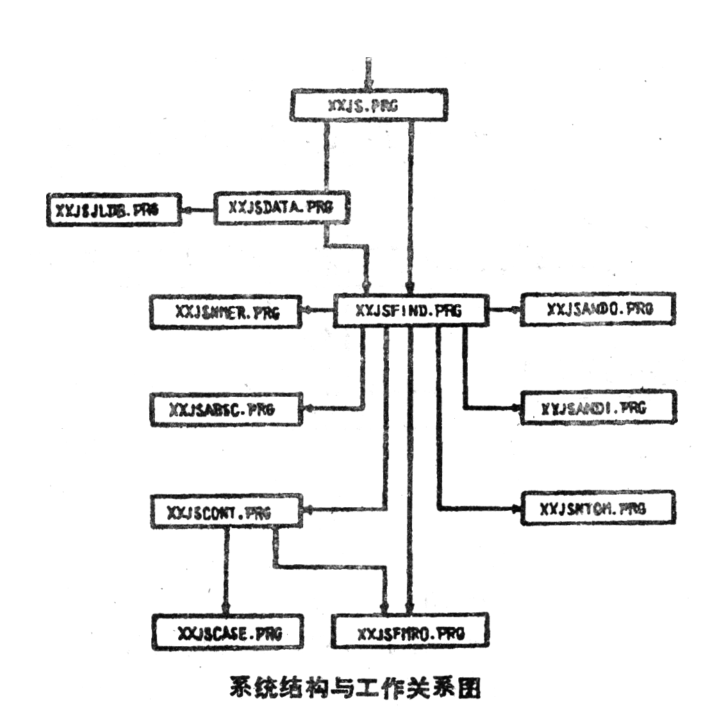
2.库维护功能。本系统对库中内容具有便利的增加、删除、更新等信息维护功能。

3.逻辑运算检索功能。本系统能以各种逻辑运算检索信息。

4.检索途径广。本系统可按主题词、作者名、出版时间及各种用户定义的叙词进行检索。

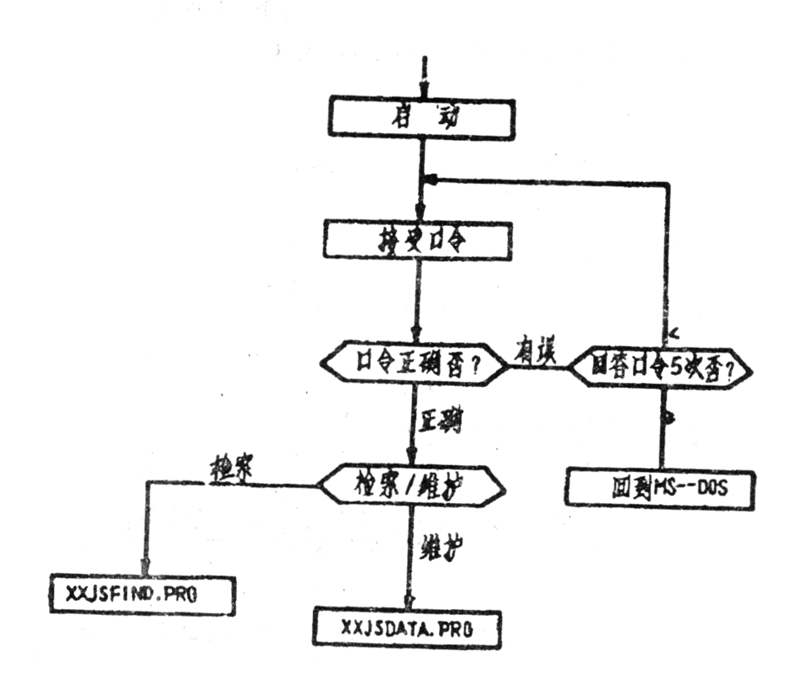
5.查错纠错功能。系统对检索者输入的错误叙词或系统没定义的叙词，具有查错并协助检索者纠错的能力。

6.本系统对检索库的超长记录具有变换存贮、跟踪检索功能。



系统结构与工作关系图

各模块流程图

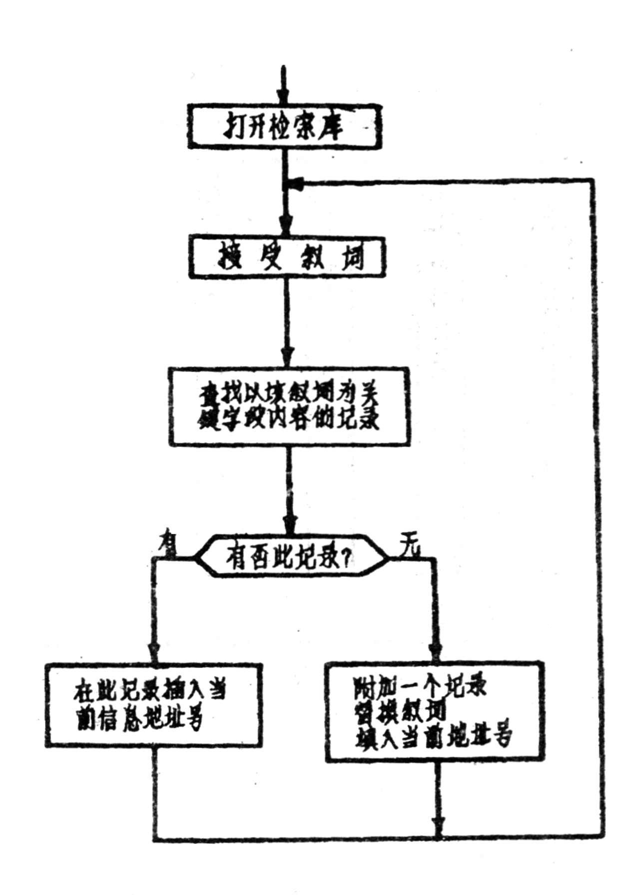


1.XXJS.PRG

功能: <1>审核用户使用权

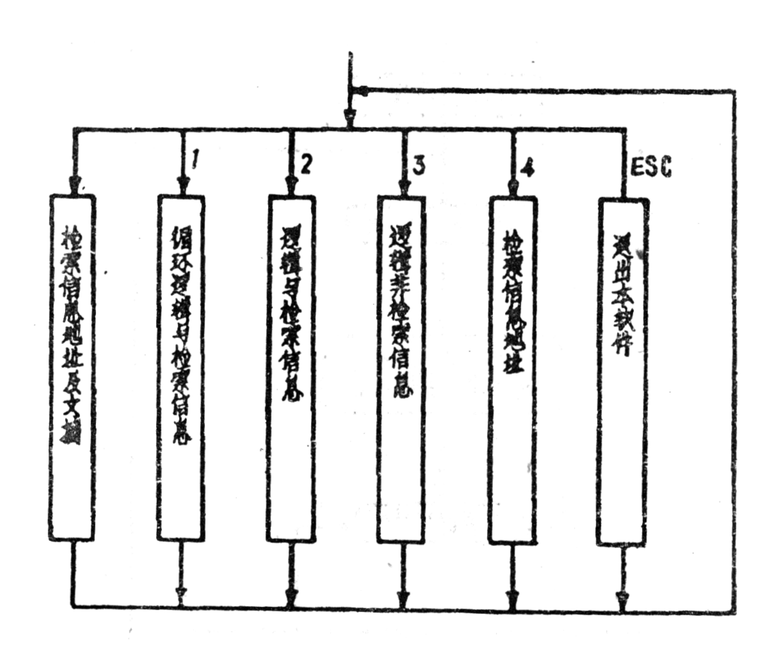
<2>调用 XXJSFIND.PRG 程序检索信息

<3>调用 XXJSDATA.PRG 程序维护库中信息



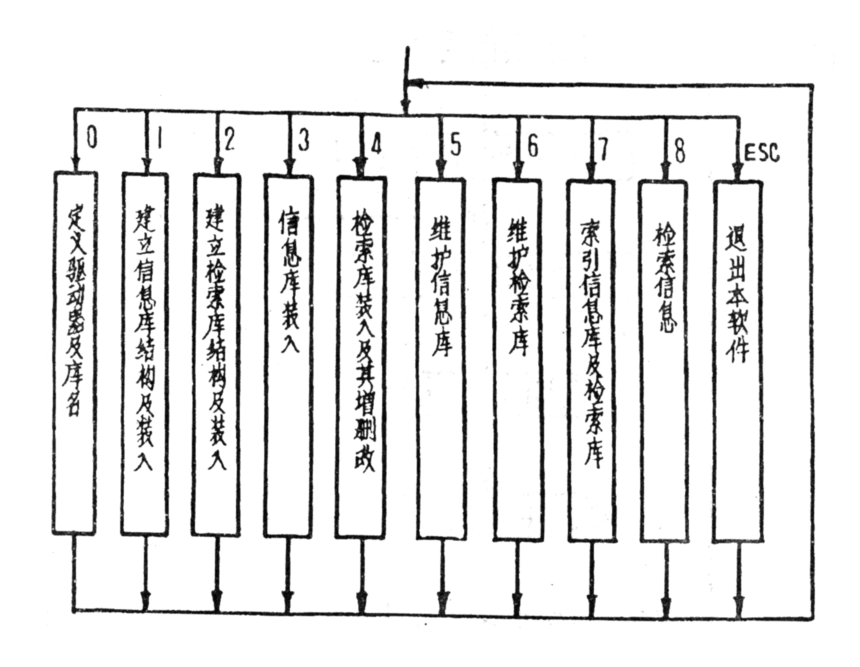
2.XXJSJLDB.PRG

功能：建立检索库



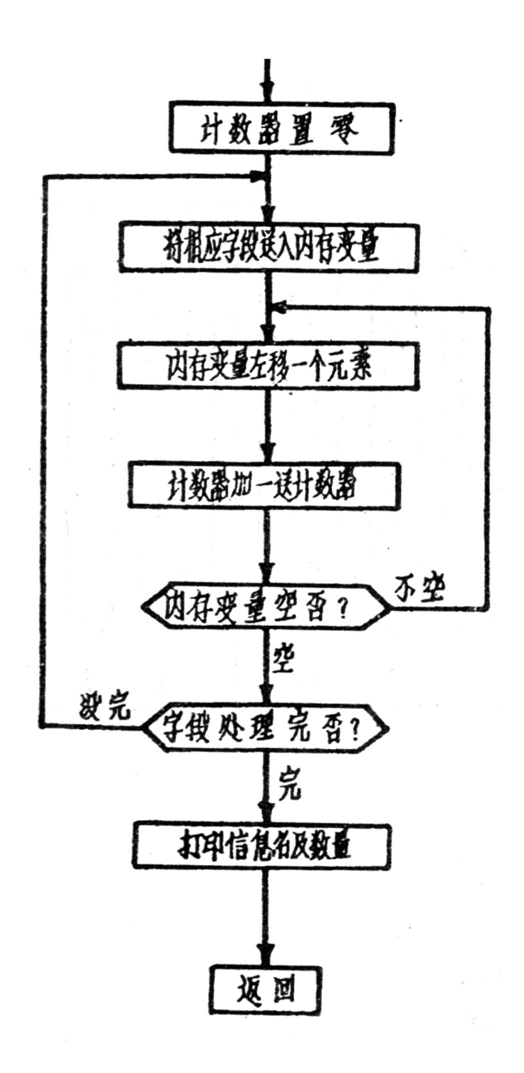
3.XXJSFIND.PRG

功能：检索信息



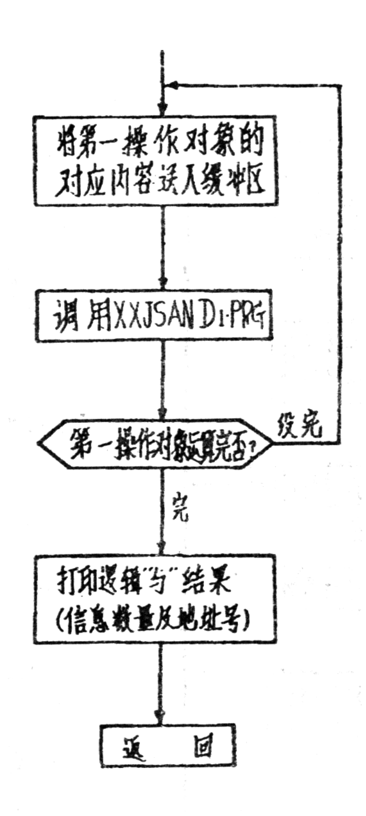
4.XXJSDATA.PRG

功能：维护库中信息



5.XXJSCONT.PRG

功能：统计信息量

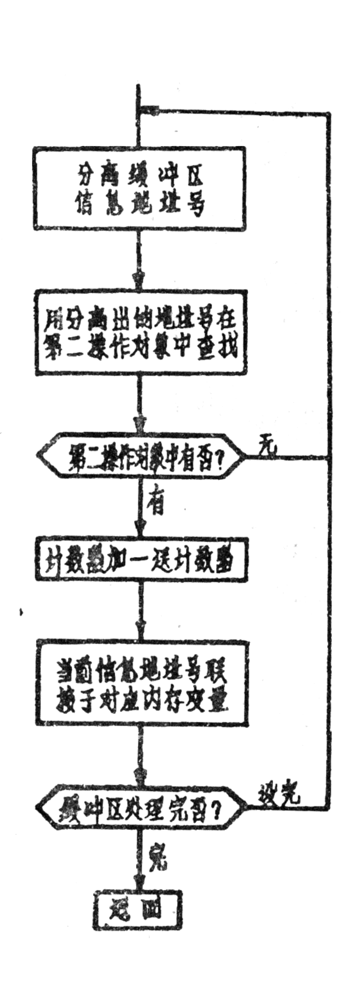


6.XXJSANDO.PRG

功能: <1>将第一操作对象的值送入缓冲区

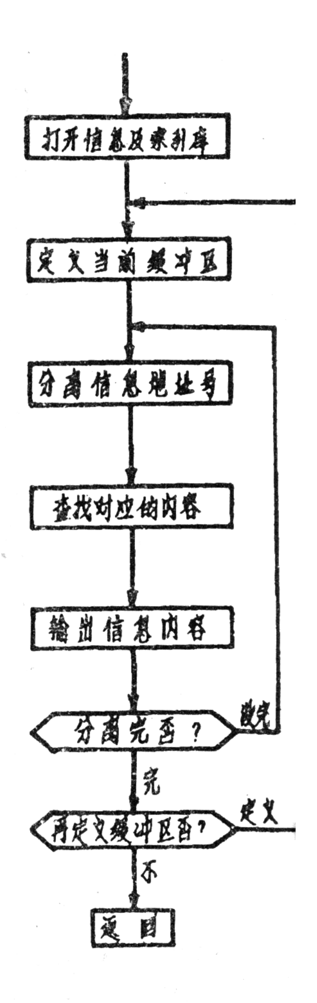
<2>调用 XXJSAND1.PRG

<3>输出逻辑"与"结果



7.XXJSAND1.PRG

功能：实现逻辑“与”运算



8.XXJSABSC.PRG

功能：检索输出信息内容

7. 4.3 源程序清单

```
* 程序名: XXJS.CMD
ERASE
RESTORE FROM XXJSMEMO.MEM
IF LOCK='N'
? 'passable'
ELSE
SET TALK OFF
STORE 0 TO M0
DO WHILE M0<5
REMARK      请输入口令!
SET CONSOLE OFF
ACCEPT 'entre the password' TO M1
SET CONSOLE ON
IF M1=LOCK
STORE 5 TO M0
ELSE
? '口令有误'
STORE M0+1 TO M0
ENDIF
IF M1<>LOCK
? '全是有误的口令，退至操作系统'
QUIT
ENDIF
ENDIF
ERASE
REMARK-------------------
```

```
?
REMARK    欢迎使用 XXJS 第 2.0 版
?
REMARK    欢迎      您对本软件提出各种意见
?
?
REMARK
REMARK————————————————
?
REMARK
REMARK
?
?
DO  WHILE  M0<100
STORE  M0+1  TO  M0
ENDDO
ERASE
? '————————————————————'
?
REMARK    键入<ENTER>键，为您检索信息
?
REMARK    键入<口令>为您维护信息库及检索库
?
? '——————————————————————'
?
? '请选择!'
?
```

```
RESTORE FROM XXJSMEMQ.MEM
SET CONSOLE OFF
ACCEPT TO CHOICE
SET CONSOLE ON
ERASE
SET TALK OFF
IF CHOICE ='DATE'
DO XXJSDATA
ELSE
DO XXJSFIND
ENDIF
RETURN
*程序名: XXJSDATA.CMD
USE
RESTORE FROM XXJSMEMQ.MEM
DO WHILE T
ERASE
?'————————————————————————'
?
?'  键入<0>，定义驱动器及库名等'
?'  键入<1>，建立信息库结构及装入'
?'  键入<2>，建立检索装入及其增、删、改'
?'  键入<3>，信息库装入'
?'  键入<4>，检索库装入及其增、删、改'
?'  键入<5>，维护信息库'
?'  键入<6>，维护检索库'
?'  键入<7>，索引信息库和检索库'
```

```
? '  键入<8>，检索信息'
? '  键入<ESC>，退出本软件'
? '-----------------------------'
?
? '您选择的是数字键'
WAIT  TO  CHOICE
DO  CASE
ERASE
? '您选择的是：定义驱动器及库名等'
?
ACCEPT  '请键入驱动器号及信息库名'  TO  FN2
?
ACCEPT  '请键入驱动器号及信息库索引文件名'
TO  FN4
?
ACCEPT  '请键入驱动器号及检索库名'  TO  FN1
?
ACCEPT  '请键入驱动器号及检索库索引文件名'
TO  FN3
?
INPUT  '请键入检索库<NUMBERi>字段宽度'
TO  M3
SAVE  TO  XXJSMEMQ.MEM
CASE  CHOICE='1'
ERASE
? '您选择的是建立信息库结构及装入'
?
```

```
? '信息库结构准备好了吗？请键入<Y>或<N>'
WAIT  TO  M6
IF  M6='Y'
?
? '请使用<<INFORNMBER>>为第1个字段名，
类型为"C"'
CREATE  &FN2
ENDIF
CASE  CHOICE='2'
ERASE
? '您选择的是，建立检索库结构及装入'
?
? '您的检索库结构设计好了吗？请键入<Y>或
<N>
WAIT  TO  M6
IF  M6='Y'
?
?'请使用<<KEY>>为第1个字段名，
<<NUMBERi>>为第i+1个字段名（i+1，2，
3，4)，类型为"C"'
CREATE  &FN1
ENDIF
CASE  CHOICE='3'
ERASE
? '您选择的是，为信息库装入信息'
USE  &FN2
APPEND
```

```
CASE  CHOICE='4'
ERASE
? '您选择的是，为检索库装入检索词及信息地址'
DO  XXJSJLDB
CASE  CHOICE='5'
ERASE
? "您选择的是，为检索库装入检索词及信息地址'
DO  XXJSJLDB
CASE  CHOICE='5'
ERASE
? '您选择的是，维护信息库'
?
USE  &FN2
USE  &FN2  INDEX  &FN4
STORE  'Y'  TO  M6
ERASE
? '下面是修改、添加、删除信息内容'
?
? '您要修改、添加、删除信息内容吗？请键入<Y>
或<N>
WAIT  TO  M6
ERASE
DO  WHILE  M6='Y
IF  M6<>'Y'
LOOP
ENDIF
ACCEPT  '请键入信息地址'  TO  MR1
```

```
FIND  &MR1
IF  INFORNMBER=MR1
?  #
EDIT
ERASE
?'您要继续修改、添加、删除信息内容吗？请键入
<Y>或<N>'
WAIT  TO  M6
ELSE
?'抱歉！ 这里不用<<&MR1>>'
?
ENDIF
ENDDO
STORE  'Y'  TO  M6
ERASE
?'您选择的是恢复带（*）号的信息记录'
?
?'您要恢复带（*）号的信息记录吗？请键入<Y>
或<N>!'
WAIT  TO  M6
ERASE
DO  WHILE  M6='Y'
IF  M6<>'Y'
LOOP
ENDIF
ACCEPT  '请键入带（*）号的信息地址'
TO  MR1
```

```
FIND  &MR1
IF  INFORNMBER=MR1
RECALL  NEXT  1
DISP
? '您要继续恢复带（*）号的信息内容吗？请键入
<Y>或<N>!'
WAIT  TO  M6
ELSE
? '抱歉！ 这里不用<<&MR1>>'
?
ENDIF
ENDDO
ERASE
? '下面供您选择的是，用<<PACK>>命令删除带
（*）号的信息内容'
?
? '注意！ 一但使用<<PACK>>命令所有带（*）
号的信息内容将永远被删除'
?
?'您一定要删除带（*）号的信息内容吗？请键入
<Y>或<N>!'
WAIT  TO  M6
IF  M6='Y'
PACK
ENDIF
CASE  CHOICE='6'
ERASE
```

```
? '您选择的是维护检索库'
USE  &FN1
USE  &FN1  INDEX  &FN3
STORE  'Y'  TO  M6
ERASE
? '下面供您选择的是，恢复带（*）号的检索词'
?
? '您要恢复带（*）号的检索词吗？请键入<Y>或
<N>!'
WAIT  TO  M6
ERASE
DO  WHILE  M6='Y'
IF  M6<>'Y'
LOOP
ENDIF
ACCEPT  '请键入希望恢复的检索词'  TO  MR1
FIND  &MR1
IF  KEY=MR1
RECALL  NEXT  1
DISP
? '您要继续恢复带（*）号的检索词吗？请键入
<Y>或<N>!'
WAIT  TO  M6
ELSE
? '抱歉！这里不用<<&MR1>>'
?
ENDIF
```

```
ENDDO
ERASE
? '下面供您选择的是，用<PACK>命令删除带
(*) 号的检索词
? '注意！ 一旦用 PACK 命令，所带 (*) 号的检索词
被删除！'
?
? '您一定要删除所有带 (*) 号的检索词吗？请键入
<Y>或<N>!
WAIT  TO  M6
IF  M6='Y'
PACK
ENDIF
CASE  CHOICE='7'
ERASE
? '您选择的是，对信息库及检索库索引'
?
? '索引信息库吗？请键入<Y>或<N>！'
WAIT  TO  M6
IF  M6='Y'
? '正在索引信息库，请等待！'
USE  &FN2
INDEX  ON  INFORNMBER  TO  &FN4
ENDIF
? '索引检索库吗？请键入<Y>或<N>1'
WAIT  TO  M6
IF  M6='Y'
```

```
? '正在索引检索库，请等待!'
USE   &FN1
INDEX   ON   KEY   TO   &FN3
ENDIF
CASE   CHOICE='8'
? '您选择的是，检索信息'
DO   XXJSFIND
ENDCASE
ENDDO
RETURN
```

```
*程序名：XXJSJLDB.CMD
USE   &FN1
SET   TALK   OFF
DO   WHILE   6912<8016
ACCEPT   '请键入检索词1'   TO   MR1
ACCEPT   '请键入检索词2'   TO   MR2
ACCEPT   '请键入检索词3'   TO   MR3
ACCEPT   '请键入检索词4'   TO   MR4
ACCEPT   '请键入检索词5'   TO   MR5
STORE   1   TO   M2
DO   WHILE   M2<6
LOCATE   NEXT   1000   FOR   KEY=MR1.OR.
KEY=MR2.OR.KEY=MR3.OR.KEY=MR4.OR.
KEY=MR5
IF   KEY=MR1.OR.KEY=MR2.OR.KEY=MR3.
```

```
OR.KEY=MR4.OR.KEY=MR5
DO  CASE
CASE  KEY=MR1
STORE  ' '  TO  MR1
CASE  KEY=MR2
STORE  ' '  TO  MR2
CASE  KEY=MR3
STOTRE  ' '  TO  MR3
CASE  KEY=MR4
STOTRE  ' '  TO  MR4
CASE  KEY=MR5
STOTRE  ' '  TO  MR5
ENDCASE
?  #
EDIT
ENDIF
STORE  M2+1  TO  M2
? MR1, MR2, MR3, MR4, MR5
ENDDO
GO  BOTTOM
STORE  1  TO  M9
DO  WHILE  M9<6
DO  CASE
CASE  MR1<>  ' '
APPEND  BLANK
REPLACE  NEXT  1  KEY  WITH  MR1
?  #
```

```
EDIT
STORE ‘  ’ TO MR1
CASE MR2<>‘  ’
APPEND BLANK
REPLACE NEXT 1 KEY WITH MR2
? #
EDIT
STORE ‘  ’ TO MR2
CASE MR3<>‘  ’
APPEND BLANK
REPLACE NEXT 1 KEY WITH MR3
? #
EDIT
STORE ‘  ’ TO MR3
CASE MR4<>‘  ’
APPEND BLANK
REPLACE NEXT 1 KEY WITH MR4
? #
EDIT
STORE ‘  ’ TO MR4
CASE MR5<>‘  ’
APPEND BLANK
REPLACE NEXT 1 KEY WITH MR5
? #
EDIT
STORE ‘  ’ TO MR5
ENDCASE
```

```
GOTO  TOP
ENDDO
RETURN

* 程序名：XXJSFIND.CMD
USE  &FN1
USE  &FN1  INDEX  &FN3
DO  WHILE  2<3
ERASE
? '     * * * 开始一个新检索过程 * * * '
?
? '———————————————————————'
?
? '       键入<0>，为您检索信息地址与文摘等'
?
? '       键入<1>，以循环逻辑"与"方式检索信息'
?
? '       键入<2>，以逻辑"与"方式检索信息'
?
? '       键入<3>，以逻辑"非"方式检索信息'
?
? '       键入<4>，为您检索信息地址'
?
? '       键入<ESC>，退出本软件'
? '———————————————————————'
?
? '您选择的是一个数字键'
```

```
?
WAIT  TO  CHOICE
ERASE
DO  CASE
CASE  CHOECE='0'
SET  PRINT  ON
?'您选择的是，检索信息地址及文摘等:'
?
SET  PRINT  OFF
STORE 'NO  WORD'  TO  M6
DO  WHILE  M6='NO  WORD'
ACCEPT  '请键入一个检索词!'  TO  MR0
STORE  MR0  TO  MR1
FIND  &MR0
IF  KEY=MR0
DO  XXJSCONT
IF  M4>0
STORE  MR1  TO  MR0
FIND  &MR0
STORE  MR0-'1'  TO  MR0
STORE  -1  TO  M7
SET  PRINT  ON
?'    <<&MR1>>信息名及文摘等:'
SET  PRINT  OFF
DO  WHILE  M7=-1  .OR.  TRIM (KEY)
=MR0
DO  XXJSNMER
```

```
DO  XXJSABSC
DO  XXJSFMR0
ENDDO
ENDIF
STORE  ' '  TO  M6
ELSE
?
?'抱歉! 这里不用<<&MR0>>'
?
STORE  'NO  WORD'  TO  M6
ENDIF
ENDDO
CASE  CHOICE='1'
STORE  0  TO  T
SET  PRINT  ON
?'您选择的是, 循环逻辑"与"检索方式:'
?
SET  PRINT  OFF
STORE  'NO  WORD'  TO  M6
DO  WHILE  M6='NO  WORD'
ACCEPT  '请输入第一个检索词!'  TO  MR1
FIND  &MR1
?
*DO  XXJSCONT
IF  KEY=MR1
DO  XXJSNTOM
DO  WHILE  M6='NO  WORD'
```

```
ACCEPT '请键入第二个检索词!  TO MR
FIND &MR
IF KEY=MR
SET PRINT ON
?
? '    <<&MR1>>"与"<<MR>>'
SET PRINT OFF
*DO XXJSCONT
STORE 0 TO M4
STORE ' ' TO MER2, MER3, MER4,
MER1
DO XXJSAND0
IF M4>0
? '您要打印信息吗? 请键入<Y>或<N>!'
WAIT TO M6
IF M6='Y'
DO XXJSABSC
ENDIF
ENDIF
STORE ' ' TO M6
ELSE
?
? '抱歉! 这里不用<<&MR>>'
?
STORE 'NO WORD' TO M6
ENDIF
ENDDO
```

```
DO  WHILE  MR<>'N'
?
ACCEPT  '您希望再作一次逻辑"与"检索吗？请键入
一个检索词或<N>!'  TO  MR
IF  MR<>'N'
FIND  &MR
IF  KEY=MR
SET  PRINT  ON
?
? '……"与"<<&MR>>'
?
SET  PRINT  OFF
IF  M4>0
STORE  MER1  TO  MNUMBER1
STORE  MER2  TO  MNUMBER2
STORE  MER3  TO  MNUMBER3
STORE  MER4  TO  MNUMBER4
ENDIF
STORE  0  TO  M4
STORE  '  '  TO  MER1, MER2, MER3,
MER4
DO  XXJSAND0
IF  M4>0
? '您要打印信息吗？请键入<Y>或<N>!'
WAIT  TO  M6
IF  M6='Y'
DO  XXJSABSC
```

```
ENDIF
ENDIF
ELSE
?
? '抱歉! 这里不用<<&MR>>'
?
ENDIF
ENDIF
ENDDO
STORE ' ' TO M6
ELSE
?
? '抱歉! 这里不用<<&MR1>>'
?
SOTRE 'NO WORD' TO M6
ENDIF
ENDDO
CASE CHOICE='2'.OR.CHOICE='3'
STORE 0 TO T
IF CHOICE='2'
SET PRINT ON
? '您选择的是，逻辑"与"检索方式'
?
?
SET PRINT OFF
ELSE
STORE 1 TO T
```

```
SET PRINT ON
? '您选择的是，逻辑"非"检索方式:'
SET PRINT OFF
ENDIF
STORE 'NO WORD' TO M6
DO WHILE M6='NO WORD'
ACCEPT '请键入第一个检索词1' TO MR
FIND &MR
?
IF KEY=MR
DO WHILE M6='NO WORD'
ACCEPT '请键入第二个检索词!' TO MR1
FIND &MR1
IF KEY=MR1
SET PRINT ON
?
IF T=1
? '<<&MR>>"非"<<&MR1>>'
ELSE
? '<<&MR>>"与"<<&MR1>>'
ENDIF
SET PRINT OFF
STORE 0 TO M4
STORE ' ' TO MER1, MER2, MER3,
MER4
STORE MR-'1' TO MR2
STORE -1 TO M8
```

```
FIND  &MR
DO  WHILE  M8=-1  .OR.  TRIM (KEY)
=MR2
STORE  MR1-'1'  TO  MR0
STORE  -1  TO  M7
DO  XXJSNTOM
FIND  &MR1
DO  WHILE  M7=-1  .OR.  TRIM (KEY)
=MR0
DO  XXJSAND0
DO  XXJSFMR0
ENDDO
STORE  M8+1  TO  M8
IF  M8<10
STORE  $ (MR2, 1, LEN (MR2) -1)
-STR (M8, 1)   TO  MR2
ELSE
IF  M8=10
STORE  $ (MR2, 1, LEN (MR2) -1)
-STR (M8, 2)   TO  MR2
ELSE
STORE  $ (MR2, 1, LEN (MR2) -2)
-STR (M8, 2)   TO  MR2
ENDIF
ENDIF
FIND  &MR2
ENDDO
```

```
STORE  ‘  ’  TO  M6
ELSE
?
?‘抱歉! 这里不用<<&MR1>>’
?
STORE  ‘NO  WORD’  DO  M6
ENDIF
ENDDO
SET  PRINT  ON
?
IF  T=1
?‘    逻辑“非”数量:’, M4
ELSE
?‘    逻辑“与”数量:’, M4
ENDIF
?
?‘    信息地址:’
?
? MER1, MER2, MER3, MER4
?
SET  PRINT  OFF
IF  M4>0
?‘您要打印信息吗? 请键入<Y>或<N>!’
WAIT  TO  M6
IF  M6=‘Y’
DO  XXJSABSC
ENDIF
```

```
ENDIF
DO  WHILE  MR<>'N'
IF  T=1
?
ACCEPT  '您希望再做一次逻辑"非"检索吗？请键入一个检索词或<N>！'  TO  MR
ELSE
?
ACCEPT  '您希望再作一次逻辑"与"检索吗？请键入一个检索词或<N>！'  TO  MR
ENDIF
IF  MR<>'N'
FIND  &MR
IF  KEY=MR
?
SET  PRINT  ON
IF  T=1
? '……"非"<<&MR>>'
ELSE
? '……"与"<<&MR>>
ENDIF
?
SET  PRINT  OFF
IF  M4>0
STORE  MER1  MNUMBER1
STORE  MER2  MNUMBER2
STORE  MER3  MNUMBER3
```

```
STORE  MER4  MNUMBER4
ENDIF
STORE  MR-'1'  TO  MR0
STORE  -1  TO  M7
STORE  0  TO  M4
STORE  '  '  TO  MER1, MER2, MER3,
MER4
DO  WHILE  M7=-1  .OR.  TRIM (KEY)
=MR0
DO  XXJSAND0
DO  XXJSFMR0
ENDDO
SET  PRINT  ON
?
IF  T=1
? '      逻辑"非"数量:', M4
ELSE
? '      逻辑"与"数量:', M4
ENDIF
?
? '      信息地址:'
?
?  MER1, MER2, MER3, MER4
?
SET  PRINT  OFF
IF  M4>0
? '您要打印信息吗? 请键入<Y>或<N>!'
```

```
WAIT TO M6
IF M6='Y'
DO XXJSABSC
ENDIF
ENDIF
ELSE
?
?'抱歉! 这里不用<<&MR>>'
?
ENDIF
ENDIF
ENDDO
STORE ' ' TO M6
ELSE
?
?'抱歉! 这里不用<<&MR>>'
?
STORE 'NO WORD' TO M6
ENDIF
ENDDO
CASE CHOICE='4'
SET PRINT ON
?'您选择的是，只检索信息地址:'
SET PRINT OFF
?
STORE 'NO WORD' TO M6
DO WHILE M6='NO WORD'
```

```
ACCEPT '请键入一个检索词' TO MR0
FIND &MR0
?
IF KEY=MR0
DO XXJSCONT
STORE ' ' TO M6
ELSE
?
?'抱歉！ 这里不用<<&MR0>>'
?
ENDIF
ENDDO
ENDCASE
ERASE
?
?'    ***本检索过程结束***'
?
DO WHILE M0<80
STORE M0+1 TO M0
ENDDO
ENDDO
RETURN
*程序名：XXJSAND0.CMD
STORE 1 TO M2
DO WHILE M2<5
DO CASE
CASE M2=1
```

```
STORE  MNUMBER1  TO  MNUMBER
CASE  M2=2
STORE  MNUMBER2  TO  MNUMBER
CASE  M2=3
STORE  MNUMBER3  TO  MNUMBER
OTHERWISE
STORE  MNUMBER4  TO  MNUMBER
ENDCASE
DO  XXJSAND1
STORE  M2+1  TO  M2
ENDDO
SET  PRINT  ON
IF  CHOICE<>'2'.AND.CHOICE<>'3'
?
? '      逻辑"与"数量:', M4
?
? '      信息地址:'
?
? MER1
? MER2
? MER3
? MER4
? '        '
ENDIF
SET  PRINT  OFF
RETURN
```

```
* 程序名: XXJSAND1.CMD
STORE  1  TO  M0
DO  WHILE  M0>0
STORE  @(',', MNUMBER)   TO  M0
IF  M0>0
STORE  $ (MNUMBER, 1, M0)   TO  M1
STORE  $ (MNUMBER, M0+1, M3)
TO  MNUMBER
STORE  @ (M1, NUMBER1) +@ (M1,
NUMBER2)  TO M5
STORE  @ (M1, NUMBER3) +@ (M1,
NUMBER4) +M5 TO M5
IF  M5>0.AND.T=0
STORE  M4+1  TO  M4
DO  CASE
CASE  M2=1.AND.LEN (MER1-M1) <M3-1
STORE  MER1-M1  TO  MER1
CASE  M2=2.AND.LEN (MER2-M1) <M3-1
STORE  MER2-M1  TO  MER2
CASE  M2=3.AND.LEN (MER3-M1) <M3-1
STORE  MER3-M1  TO  MER2
OTHERWISE
IF  LEN (MER4-M1) <M3-1
STORE  MER4-M1  TO  MER4
ENDIF
ENDCASE
ELSE
```

```
IF  M5=0.AND.T=1
STORE  M4+1  TO  M4
ENDIF
DO  CASE
CASE  M2=1.AND.LEN (MER1-M1) <M3-1
STORE  MER1-M1  TO  MER1
CASE  M2=2.AND.LEN (MER2-M1) <M3-1
STORE  MER2-M1  TO  MER2
CASE  M2=3.AND.LEN (MER3-M1) <M3-1
STORE  MER3-M1  TO  MER3
OTHERWISE
IF  LEN (MER4-M1) <M3-1
STORE  MER4-M1  TO  MER4
ENDIF
ENDCASE
ENDIF
ENDIF
ENDIF
ENDDO
RETURN

*程序名：XXJSABSC.CMD
STORE  1  TO  M2
SET  PRINT  ON
IF  CHOICE  <>'0'
?
?'  信息名及文摘等:'
```

```
?
ENDIF
SET  PRINT  OFF
USE  &FN2
USE  &FN2  INDEX  &FN4
DO  WHILE  M2<5
DO  CASE
CASE  M2=1
STORE  MER1  TO  MNUMBER
CASE  M2=2
STORE  MER2  TO  MNUMBER
CASE  M2=3
STORE  MER3  TO  MNUMBER
OTHERWISE
STORE  MER4  TO  MNUMBER
ENDCASE
STORE  1  TO  M0
DO  WHILE  M0>0
STORE  @(',', MNUMBER  TO  M0
IF  M0>0
STORE  $ (MNUMBER, 1, M0-1)   TO  M1
STORE  $ (MNUMBER, M0+1, M3)
TO  MNUMBER
FIND  &M1
SET  PRINT  ON
*INFORNMBER, INFORNAME, 作者, 出版
社, 时间, 单价
```

```
DISP  OFF
SET  PRINT  OFF
ENDIF
ENDDO
STORE  M2+1  TO  M2
ENDDO
USE  &FN1
USE  &FN1  INDEX  &FN3
RETURN

*程序名: XXJSCONT.CMD
STORE  0  TO  M4
SET  PRINT  ON
?
? '      <<&MR0>>信息地址:'
?
STORE  MR0-'1'  TO  MR0
STORE  -1  TO  M7
DO  WHILE  M7=-1.OR.TRIM (KEY) =MR0
STORE  1  TO  M2
DO  WHILE  M2<5
DO  XXJSCASE
STORE  1  TO  M0
DO  WHILE  M0>0
STORE  @(',', MNUMBER)  TO  M0
IF  M0>0
STORE  $ (MNUMBER, M0+1, M3)
```

```
TO  MNUMBER
STORE  M4+1  TO  M4
ENDIF
ENDDO
STORE  M2+1  TO  M2
ENDDO
STORE  ‘  ’  TO  MNUMBER
SET  PRINT  ON
DISP  OFF
DO  XXJSFMR0
ENDDO
?
? ‘     信息总数:’, M4
?
SET  PRINT  OFF
RETURN
```

```
* 程序名: XXJSINIT.CMD
ACCEPT  ‘设置口令吗? 键入 Y 或 N’  TO  LOCK
IF  LOCK = ‘Y’
ACCEPT  ‘键入口令’  TO  LOCK
ENDIF
SAVE  TO  XXJSMEMO.MEM
RETURN
```

```
程序名: XXJSCASE.CMD
DO  CASE
```

```
CASE  M2=1
STORE  NUMBER1  TO  MNUMBER
CASE  M2=2
STORE  NUMBER2  TO  MNUMBER
CASE  M2=3
STORE  NUMBER3  TO  MNUMBER
OTHERWISE
STORE  NUMBER4  TO  MNUMBER
ENDCASE
RETURN
```

```
* 程序名：XXJSFMR0.CMD
STORE  M7+1  TO  M7
IF  M7<10
STORE  $ (MR0, 1, LEN (MR0) -1)
-STR (M7, 1)   TO  MR0
ELSE
IF  M7=10
STORE  $ (MR0, 1, LEN (MR0) -1)
-STR (M7, 2)   TO  MR0
ELSE
STORE  $ (MR0, 1, LEN (MR0) -2)
-STR (M7, 2)  TO MR0
ENDIF
ENDIF
FIND  &MR0
RETURN
```

```
程序名：XXJSNMER.CMD
STORE   NUMBER1   TO   MER1
STORE   NUMBER2   TO   MER2
STORE   NUMBER3   TO   MER3
STORE   NUMBER4   TO   MER4
RETURN
```

```
* 程序名：XXJSNTOM.CMD
STORE   NUMBER1   TO   MNUMBER1
STORE   NUMBER2   TO   MNUMBER2
STORE   NUMBER3   TO   MNUMBER3
STORE   NUMBER4   TO   MNUMBER4
RETURN
```

习题七

1. 结合自己的学习和工作实际设计一个项目(如人事或其它)的软件，应具有信息录入、修改、插入、删除、检索(包括多种条件检索)、统计等功能。

2. 在习题1的基础上扩充对数据库内容的表格(页式或非页式)打印功能。

第八章　cdBASE 程序设计方法与技巧

8.1　程序设计概述

8.1.1　程序和程序设计

什么是程序？有人说：程序是一个指令序列。还有人说：为解决某一问题而设计的一系列指令称为程序，等等。众说不一，但意思接近。通俗地讲：为完成一项工作或任务而安排的步骤均可称为程序。例如：食谱、乐谱都是程序。每个人每天都在履行一定的程序。什么是指令呢？从字义上讲指令可以解释为指示命令。对于食谱来说，加一匙食盐，放几粒花椒都是指令。对计算机指令可以理解为，说明计算机执行一种操作的一组符号。本章所述的程序一般是指计算机系统运行的程序，对于计算机这个专业领域来讲，告诉计算机如何做某一事情的一组指令序列称为程序。它可以用计算机指令或程序设计语言来编写。

什么是程序设计？简而言之，设计程序的过程称为程序设计。程序设计和程序的概念一样，并不仅是和计算机联在一起。有烹调经验的人为生日家宴制作食谱，作曲家研制

“乐谱”的过程都可看成是程序设计。当然对于计算机领域内所指的程序设计，是有其专业内函的，本章所谈的程序设计是指正确、有效地组织计算机依次执行的每一条指令或语句。

程序设计是一项技术性技巧性很强的工作，是模仿性和创造性的统一。所谓模仿性就是在进行程序设计时，所用的指令或语句，变量格式等都必须严格符合程序的语言所规定的语法。所谓创造性就是在符合程序设计语言所规定的语法前提下，程序设计者可以针对各自问题尽情发挥。

程序设计是逻辑思维。由于计算机本身是没有创造能力和普通生物的感觉器官的机器，必须向它提供清晰而无二义性的指令。所以它执行程序是严格按照指令的含义执行，不是对，就是错，或者说，不是真就是假。这就要求程序设计者遵重计算机的特性，采用逻辑思维的方法。

程序设计既复杂，又灵活，没有一种统一模式，程序设计人员可以尽情发挥。和写文章一样，一个对某种自然语言（如汉语）掌握得很好的人，不一定就能写出好文章，一个对计算机语句，命令学得很好的人，也不等于能设计出好的程序。对于同一个题目，十个人写的文章会十个样子；十个人对同一课题设计的程序也不尽相同，原因在于二者都是手工生产。程序设计不仅要满足人的要求而且还要符合机器的要求，由此说来，程序设计是一项极其复杂又十分灵活的工作。

但是，随着计算机科学技术的发展，计算机对于社会各个行业（包括人们的日常生活）产生的深远影响，将促使个人计算机及其被称为个人程序设计时代的到来。需要普及程序设计的基础知识和基本技能。

8.1.2 程序设计的基本技术

本节介绍程序设计的一些基本手段、原则、方法、步骤等基本技术。

1. 分解

应付复杂问题的一个重要手段是“分解”，即“分而治之”。将程序设计工作划分成几个阶段，把一个程序模块化等都体现了“分解”。分解的含义是将一个大问题分割成若干个较小的、易于解决的部分，然后分别处理。

随意机械的分解有可能使问题更加复杂化。在弄清问题内部存在的各种关系或联系机制后，科学地分解，可使问题简化且易于处理。程序设计中的分解应使各个阶段之间的联系尽量少，各个模块间的联系尽量少。

2. 抽象

程序设计的基本原则之一是抽象，抽象是应付复杂问题的又一手段，人的智力往往不可能一下子涉及到问题的全部细节。合理的采用抽象可使问题简化、易于理解，自顶向下逐层分解，上一层是下一层的抽象，将大量细节放在下一层，使上一层得到简化。抽象偏重于反映问题的本质属性，一般抽象在前，抽象阶段着重考虑“做什么”。而在实现阶段着重考虑“如何做”，把一个关系复杂、规模庞大、系统性强的问题一步一步一层一层地加以解决。

3. 逐步求精

逐步求精是一种适应性强并广泛采用的程序设计方法。逐步求精是建立在抽象原则的基础上的，经过抽象产生抽象的模型、数据及程序。这些抽象的模型，数据及程序是不完善的，需要通过逐步求精的方法一步一步的完善，一直到最

后程序才能满足应用实际的需要。

4. 确定性

程序设计的又一原则是确定性。确定性是保证程序质量的前提，开初对问题的一些含糊的直觉，要逐步变成明确的描述，使其确定化，例如："程序质量好"要具体化成"易维护、可靠、高效、易理解"等；"模块质量好"具体化为"块内联系多，块间联系少"，象食谱中放一匙食盐这种不十分确定的指令描述，在计算机执行的程序中是不能接受的。

5. 一致性

一致性表现是多方面的，在一个程序中，设计风格、变量或子程序等命名方式，都应保持一致性。在程序设计之初对所用文件名，变量名要有一个大概的划分，尽量有规律性！统一性，这样做可以使程序清晰，避免引起变量或文件名的重复、混乱，有利于程序的调试；文件的拷贝，列目录等工作。

统一化、标准化是程序成功的关键。讨论程序设计方法的目的之一，使程序设计走上标准化的轨道，这样作对程序的易读、易维护等质量，及效率、生命周期等都将产生重大的影响。

6. 程序设计步骤

程序设计不仅对有关命令的功能要有深透的理解，更重要的是如何结合具体问题去组织这些命令。对于复杂的问题，程序设计必须分步进行。比较易于掌握且行之有效的是程序设计五步法。即：

第一步：审题。主要是分析弄清要解决什么问题。有多大的数据信息量，对内存、外存、运行速度的要求如何？经费时间、技术、设备、人才等条件是否具备，用计算机解决

本问题是否可行。

第二步：画出功能模块图。根据要求和具体条件的可能，程序应具备哪些功能，按功能画模块图。图 8.1 是一个最简单的功能模块图。

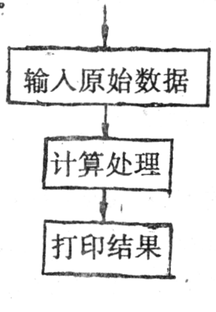


图 8.1

第三步：数据结构设计。数据的组织，直接影响解决方法及程序设计，影响到程序质量，数据结构好，可以简化程序设计。数据的结构还会影响到存储量、处理速度、输入／输出次数、数据的冗余度、程序的可靠性等。

对于传统计算问题来讲，数据是单个的还是成组的，是数组还是文件？是随机文件还是顺序文件？等等。对于信息管理来说，数据库的数量，各个数据库的字段名，宽度，类型等等。数据结构的理论与技术比较复杂，可参看其他数据结构方面的著作。

第四步：画出程序流程图。在分析问题、确认功能模块图、设计数据结构的基础上，画出完成各功能模块的具体流程图。这样的流程图更能满足编程实现的需要，考虑的问题也比较深入，描绘了“如何做”的问题。

第五步：编写程序。编写程序时，要根据第四步中的程

序流程图，结合具体问题的要求，对照一些有关的实例进行。对一些与机器关系密切又无把握的地方要上机试验。最后全部通过。

以上的五步，要经反复修改，逐步求精。才能得到一个正确的程序。

8.1.3 结构程序设计概述

1. 结构程序设计

美国 IBM 公司的 W·steven 和 G·myers 等人提出的结构化设计（Structured Design）方法，是使用最广的程序设计方法。它运用“分解”控制程序设计工作的复杂性。“分解”就是将程序划分成一个个模块，即模块化。把一个复杂的程序设计成相对独立、功能单一的模块结构后，就可单独地分工进行设计和编程，这样能够提高程序的质量并简化程序设计工作。

评价模块结构的质量标准是，块间联系最小，块内联系最大。结构程序设计的总则是使每个模块完成一个功能，模块间的传送参数尽量少。

结构程序设计首先要考虑的问题有：如何将程序化分成相对独立的模块；模块间传送一些什么信息；模块间的调用关系如何。

什么是模块，模块是指用一个名字可以调用的一段程序，它的概念类似于子程序。在设计的时候，一般是指这些程序模块的原形，或称功能模块。

2. 结构图

结构设计的描述方式是结构图，它反映了程序的模块结构、块间联系以及块内联系等特性。结构图中的主要成分

有:

模块——用方框表示，方框中标明反映模块功能的名字。

调用——用箭头表示模块间的调用关系，箭头的尾部模块称为调用模块，箭头指向的模块称为被调用模块。

数据——在调用箭头旁边用小箭头及字母等表示一个模块调用另一模块时的传送数据。

条件调用——在箭头的尾部用菱形表示有条件的调用。

循环调用——在箭头的尾部用弧形箭头表示循环调用。

上述符号的使用如图 8.2 所示。

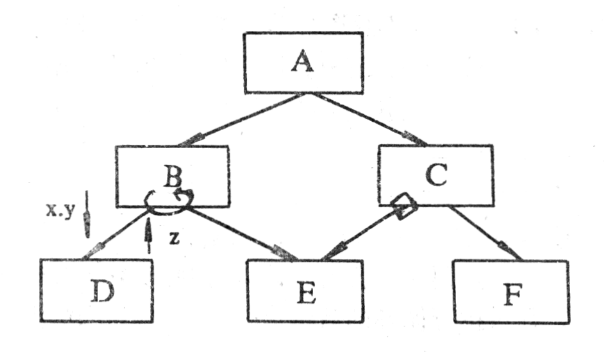


图 *8.2* 结构图

图中模块 A 调用模块 B 和模块 C；模块 B 循环调用模块 D 和 E，调用模块 D 时传送数据 X，Y，返回数据 Z；模块 C 有条件地调用模块 E，无条件调用模块 F。

3.　程序设计语言的语句结构

当今的程序设计语言不甚枚举，各种机器都配上了多种

语言。纵观这些流行且完善的语言，其程序的执行顺序都是由语句结构来决定的，我们在阅读他人程序或自己编程时应当有意留心结构的组成方式，最常见的语句结构如下:

①顺序结构

顺序结构是指在程序运行时严格按语句排列的先后顺序一条一条地执行。这种结构简单明了，但不能描述某些复杂的问题。

②判断结构

判断结构是由语句中的判断条件来决定程序执行的走向。判断结构又分为几种:①如果〈条件满足〉则〈执行语句序列〉;②如果〈条件满足〉则〈执行语句序列 1〉否则〈执行语句序列 2〉;③多情形判断。如:〈满足条件 1〉则〈执行语句序列 1〉;〈满足条件 2〉则〈执行语句序列 2〉;…〈满足条件 n〉则〈执行语句序列 n〉;〈其它情形〉则〈执行语句序列 n+1〉。

③循环结构

循环结构就是按给定条件去重复执行一段程序。常用的是:满足〈条件〉则〈执行循环体〉，如此反复，直到不满足〈条件〉则〈执行循环的后继语句〉。循环结构能大大缩短程序的长度。

④调用结构

循环结构解决的是一个语句序列就地多次执行的问题，调用结构解决的是一个语句序列在不同的多个地方多次执行的问题，一般被调用的语句序列是一个独立的模块。图 8.2 中的 B、C、D、E、F 均是被调用的模块。

8.1.4 程序质量

讨论程序设计技术的目的是提高程序质量。早期，注重强调程序的正确性和效率。随着计算机的普及和硬件价格的大幅度下降，加之程序规模和复杂性的增加，人们对程序的质量倾向在可靠、易维护、效率、易理解等几个方面进行评价。

1. 可靠性

可靠性包括两个概念:

①正确性：指程序本身没有错误，能正确实现程序功能。

②健壮性：指在硬件发生故障或输入数据不合理等，系统仍能作出适当反映。

总而言之，可靠性是指系统在意料的情况下，能够正确地工作；在意外情况下，能作适当处理，不会造成严重损害。

2. 易维护

通常把易阅读、易发现和纠正错误，易修改扩充归结为易维护。在设计和建立文档时都精心考虑到程序的易维护。使易于排错或扩充程序功能。

3. 效率

效率是指程序要有效地发挥和使用计算机资源，节省时间和空间。在当今微机价格大幅度下降，不断普及且不存在机时紧张的情况下，只要不影响应用问题的解决，不必过多追求效率。但对那些占用机时过长，如连续运转几个小时或更长以至失去实用价值的程序，就必须千方百计提高效率，减少运行时间，对于那些似乎容量不够的情况，应尽量节省容量，达到在较小的资源环境下，运行较大的程序。

5. 易理解

易理解一方面是指程序内部结构清晰，易于软件人员阅读和理解；另一方面指程序的人——机界面简明清晰，使用方便。

以上几个目标既联系，又矛盾，要在研制经费，研制时间，可使用的资源等限制条件下，由程序人员灵活掌握。

8.2 cdBASE 应用程序设计

cdBASE 是一个数据库管理系统，它包括了叫做 dBASE 的程序设计语言。使用 dBASE 程序设计语言，可针对用户的特殊需要来开发应用程序。对于数据处理来说，用 cdBASE 开发程序，比用 COBOL 或 BASIC 等语言简便得多。对于计算机程序设计的初学者来说，用 dBASE 作为程序设计的入门，的确是一个捷径。

8.2.1 dBASE 应用程序设计的步骤

1. 分析

在开始设计 dBASE 数据库的应用程序之前，首先应考虑下列问题:

①要求程序做些什么?

②用 dBASE 设计的程序能否满足需要?

③要求程序怎样工作?

④程序是否需要联机帮助?

⑤当对程序进行扩充和改进时，是否比较容易?

⑥如何建立数据库文件或其它文件?

⑦程序是从键盘还是数据库文件获得数据?

⑧程序对输入和输出进行检验或修改吗?

⑨需要哪些报表打印和屏幕格式？

在开始设计之前，必须精确地定义出程序要做什么。从所需要的输出开始，分析输入，然后确定程序将怎样产生这些输出。例如：如果要求程序产生日、月、季和年报表，则要确定这些报表中有什么样的信息，应怎样构造并打印出这些信息。如果程序设计者不是程序的使用者，程序设计者就要与使用者一起反复商讨上述问题。这样，程序才能更好地满足实际应用的要求。

2. 设计

最常用的程序设计方法是自顶向下的结构设计方法。即从一个概括的情形开始，一步一步地工作，一直做到细节为止。在精确地定义出程序要做什么之后，用文字及图表描述整个程序设计目标，以便搞清楚它的总体结构。这时不要考虑程序的具体实现细节，重要的是确定有关的步骤。例如：要求一个程序做的事情是显示数据文件中的记录，应写出如下概要。

①建立工作环境。

②用菜单提示用户该程序可做什么。

③得到数据库文件名和任何有关的文件名，如索引文件名等。

④打开数据库文件以及有关文件。

⑤确定要显示的记录。

⑥确定要显示哪些字段。

⑦显示所需记录的所需字段。

⑧询问用户是否还要显示其它记录。

⑨若要，重复⑤到⑦

⑩若不要，关闭数据库文件并且返回。

在确定了整个程序流程之后，标识出程序要执行的基本活动。然后，逐步地把总的活动细分为越来越小的部分或模块，就象总框中的子部分一样。每个模块最终应该完成一件特定的任务。对于可能性最小的问题也不能忽视。

一旦把程序的各个任务划分成了明确部分，就可以单独地为每个部分写出程序。由于每次只编写一个基本任务，所以就能很容易地写出一个程序块。把一个完整的程序设计项目划分成一个一个的逻辑部分的方法称为结构程序设计，或称模块设计方法。

3. "半成品"程序

在完成了每个模块的分解之后，以文字和程序语句相结合的形式写出模块的目的。这种技术称之为伪编码。然后，在伪编码基础上写出源程序。这样作可使设计更精细，编码(写出源程序）更容易。

4. 注重调用标准模块

自顶向下和结构化程序设计技术，除了使得一个复杂的项目简化之外，还允许在任何需要的时候重复使用这些模块。在一个项目中可以在多个地方使用同一个模块这就是软件工程中"软件重用"的新概念。可以大大提高开发程序的效率。经验丰富的软件开发人员一般都积累有较多的程序模块，称之为工具箱或标准模块，在必要时拷贝下来，作一定改动即成。

5. 适当提供程序注释

源程序以及整个程序的有关文件资料是程序开发的重要内容。在源程序中提供程序段功能的注释也是十分有益的。在 dBASE 中用星号（*）或 NOTE 作为一个注释行的开始。注释有助于理解程序的逻辑，有利于测试时查找问题和

错误，有利于今后的扩充、修改等维护工作。

6. 程序扩充

以模块方式设计的程序非常容易扩充。例如在“菜单”中增加选择项目，不会影响到程序的其它部分。只需在“菜单”程序中加上增加选择项目及调用语句，并增加新的子程序就可以了。

7. 多次测试

写完一个模块后，应对它进行充分的检测。这样使得出现的问题在本模块内找到原因，便于排除错误。检测时应力图使程序出错，发现问题后才能解决问题。

8. 模块组装

在完成各个模块的调试之后，应逐步把各个模块在整个程序设计的意义下组装起来，进行总体调试，最后形成一个完整的程序。

9. 程序开发步骤小结

①确定程序要做什么。

②采用自顶向下方法确定程序的整个流程和结构，把程序划分成逻辑部分或模块，再继续把这些部分一直分到每个模块完成一个特定的任务。画出程序设计框图。

③分别写出每个模块，先用文字，再用伪码，最后用dBASE 语句。

④一边设计，一边整理设计项目和源程序的资料。

⑤单独测试每个模块。

⑥组装并对最终的程序作彻底的测试。

8.2.2 dBASE 程序及语句结构

1. 程序结构

一般来说，dBASE 程序应具有以下几个部分：

①程序的"前言"

程序的前言是指在程序的开头，对程序的名字、功能、作者、编辑历史开发程序的起止时间等信息给出一个说明。

②程序的运行环境设置

这部分一般跟在"前言"的后面，为程序建立操作环境。用 SET 命令来设置，一般是在调用程序中设置工作环境。如：SET TALK OFF 等，必要时也可在相应的子程序中设置。

③程序体

程序体由完成程序功能的命令构成。诸如输入、显示、修改数据库中的信息、加工处理、输出。还有一个值得注意的问题是合理的安排中断程序及调用其它程序模块。

④关闭文件和恢复环境部分

每个程序都要做一些整理工作。例如：应保证程序在返回到点状态或操作系统之前，恢复标准特征值；为保证数据的完整性应在适当的地方关闭数据库文件等等。

2. 语句结构

①顺序结构

顺序结构是指程序运行时，按先后逐个语句执行。dBASE 规定一个语句的长度不能超过 254 个字符，一个语句超过屏幕宽度（80 个字符）时，可用分号(;）分成多行。dBASE 执行时视为一个语句看待。

②判断结构

判断就是以语句中的判断条件来决定执行程序走向。

1)简单判断。其格式是：

IF〈条件〉

```
〈语句序列〉
ENDIF
```

2)选择判断。其格式是:

```
IF〈条件〉
〈语句序列 1〉
ELSE
〈语句序列 2〉
ENDIF
```

3)多重选择。其格式是:

```
IF〈条件 1〉
〈语句序列 1〉
ELSE
    IF〈条件 2〉
      〈语句序列 2〉
    ELSE
        IF〈条件 3〉
          〈语句序列 3〉
        ELSE
          ……
        ENDIF
    ENDIF
ENDIF
```

4)结构式选择判断

```
DO  CASE
     CASE〈条件 1〉
      〈语句序列 1〉
      〔CASE〈条件 2〉
```

〈语句序列 2〉

……

CASE〈条件 n〉

〈语句序列 n〉〕

〔OTHERWISE

〈语句序列 n+1〉〕

ENDCASE

其中方括号中的内容表示可有可无。

③循环

循环就是按给定条件重复执行本段程序。格式是:

DO WHILE〈条件〉

〈语句序列 1〉

〔LOOP〕

〈语句序列 2〉

ENDDO

④调用结构。

其格式是:

DO〈子程序名〉

dBASE 的方便之处在于各个程序模块都可根据需要进行调用。这种调用可以多至 16 层。这样的调用解决了多个地方多次甚至重复执行同一程序模块的问题。

8.3 cdBASE 实用技巧

本节介绍一些实用的 dBASE 数据库使用及程序设计技巧。

8.3.1 巧用数据库文件

cdBASE 数据库的设计对其程序设计的影响很大，如果能遵循数据库文件的关系模式，那么就越能体现 dBASE 的优越性。对于一个应用项目，可能要建立若干个不同的，然而是有关系的数据库文件，在不同的数据库文件之间，可以用一个字段作为桥梁把它们联系起来。

1. **由源库生成目标库**

由于 dBASE 对于字段太多或记录数太多的数据库操作速度较低，且在断电时有可能使正在应用的数据库丢失数据或造成混乱。为了提高速度及数据安全保护的需要，在对数据库进行操作时，尽可能在目标库中进行。所谓目标库是指由原始库的数据拷贝而得到的一个新库。由 dBASE 对原有的数据库产生新库十分方便。

①照源库原样产生新库

USE〈源库名〉

COPY TO〈目标库名〉

②仅复制源库的结构到目标库

USE〈源库名〉

COPY TO〈目标库名〉STRUCTURE

③仅复制源库结构中的几个字段到目的库

USE〈源库名〉

COPY TO〈目标库〉STRU〔FIELD〈字段名表〉〕

④复制部分结构连同部分数据到目的库

USE〈源库名〉

COPY TO〈目的库名〉FIELD〈字段表〉

⑤保留源库全部内容，同时需增加字段名

USE〈源库名〉

COPY TO〈目标库名〉

MODI STRU（回答 Y，即消除全部数据，用键盘命令增加字段，然后用 CTRL／W 写盘）

APPE FROM〈目标库名〉

DELE FILE〈目标库名〉

此时的源库已增加字段名

⑥保留源库数据，并改变字段名

USE〈旧库名〉

COPY TO〈暂存外部文件名〉SDF

MODI STRU（回答 Y 后，修改字段，存盘）

APPE FROM〈暂存外部文件名〉·TXT SDF

DELE FILE〈暂存外部文件名〉·TXT

⑦从间接数据库产生目的库

USE〈待仿建的源库名〉

COPY TO〈间接库名〉STRU

CREA〈目标库名〉FROM〈间接库名〉

有时新的库要求：

A 原来的数据库中字段名的顺序要重新排列

B 可能要增加几个字段（用⑤的方法）

C 可能要减少几上字段（用④的方法）

对 A，可用下法：

USE OLD（例 OLD. DBF 中顺序为：工号，基本工资，姓名……）

COPY TO TEMP STRU

USE TEMP

DISP STRU

MODI STRU（打 Y，将“基本工资”改为姓名，C，8

将“姓名”字段改为基本工资，N，6，2

……按 CTRL／W)

APPE FROM LOD

COPY TO NEW（在 NEW.DBF 中，顺序为：工号，姓名，基本工资，…)

2. 裁剪数据库

为了提高效率，根据不同的应用需要将数据库中的数据横向或纵向进行裁剪，横向裁剪就是去掉一时不用的字段，纵向裁剪就是去掉一时不用的记录，在经过裁剪之后的小数据库上操作，可以大大提高效率。裁剪是用拷贝命令来实现的，从源库中选出必要的字段和记录。如:

USE BRXX

COPY TO BRXX1 FIELD 疾病编号，疗效 FOR 疾病编号＞0。

这两条命令的作用是，从一个庞大的医院住院病人信息库中，选出供疾病分类统计用的疾病编号大于 0 的记录（假定疾病编号＝0 的住院者为产科非病人，不属统计之列）的疾病编号和疗效两个字段。

3. 合并数据库

合并数据库也包括横向合并或纵向合并，对于横向合并 dBASE 的 JOIN 命令可直接进行；纵向合并可用几条命令完成。例如：假定在某一应用项目中需要将各个下属单位软盘中的 DATA.DBF 文件合并到一个大库，以便在大库上加工获得整个大单位汇总数据。对此可用如下的程序段:

STORE ‘Y’ TO HD

USE C:DATA

DO WHILE HD＝‘Y’

? ″请将下属单位软盘插入驱动器 A:，并按回车键
WAIT
APPEND FROM A:DATA.DBF
?“继续吗？(Y/N)”
WAIT TO HD
ENDDO

执行本程序段的功能是逐一录入下属单位软盘上的数据，直到回答 N 时，表示各下属单位的数据盘已录入完毕。实现了将各下属单位的 DATA.DBF 合并到整个单位 C:盘中的 DATA.BDF 数据库文件中。再用索引或排序及 TOTAL 命令可获得整个大单位汇总数据。

8.3.2 检查键盘输入

对用户输入错误的检查，不只是在屏幕格式中，还包括在用户输入之后的程序检查，这是程序可靠性的要求；并且还应使程序具有对用户不正确选择的处理能力。

1. 检查“菜单选择”

①假定某“菜单”具有 0～9 种选择，可用如下方法限制检查。

```
STORE 1 TO XZ
@ 5, 5 GET XZ PICTURE‘9’
READ
```

如此可防止用户按字母及 0～9 以外的其它键。

②假定某“菜单”选择限制在 1～7 的范围内，则可由如下程序段实现。

```
STORE 0 TO XZ
DO WHILE XZ<1.OR.XZ>7
WAIT “请输入功能选择 (1～7)” TO XZ
```

```
ENDDO
```

本程序段拒绝接受任何一个比 1 小或比 7 大的数字，如果使用者输入了不正确的数字，则要求重新输入，直到输入为 1～7 的数字为止。

③假定某管理软件的主控程序如下：

```
STORE ' ' TO N
DO WHILE .T.
ERASE
? "* * * ×××管理系统 * * *"
? "1.数据输入      2.数据修改"
? "3.数据加工      4.数据显示"
? "5.报表打印      0.退出"
@ 8, 9 SAY "请输入功能选择号:" GET N
READ
IF N="0"
EXIT
ELSE IF N<="5"
DO PRG&N
ENDIF
ENDIF
ENDDO
```

本程序段比用多条 CASE 语句节省许多语句行。增加某些功能模块，基本上无需修改控制部分。显然，对 IF N<="5"中的 5 要随增加的功能模块变化，各功能的模块名都应由 PRG 与其序号组成。

2. 检查用户回答

在用户键入了回答之后，如何用程序检查输入的回答超

出范围与否，下面是一个让用户回答（Y／N）表示是否继续的检查程序段。

```
STORE " " TO HD
DO WHILE HD#"Y".AND.HD#"N"
@ 8, 8 SAY "还继续吗？(Y／N)" GET HD
PECTURE"!"
READ
ENDDO
```

本程序段不论输入大写N或Y，还是小写n或y，屏幕显示或程序接收都是大写。只要用户打入的不是Y或N，程序段请求重新输入。

3. 检查输入的表达式

在检索时只有输入合法的条件表达式才能进行，在自由的随机条件检索（即用户临时决定检索条件）情况下，更有必要。

```
STORE "                    " TO SR
DO WHILE TYPE(SR)="C".OR.TYPE(SR)="N"
ACCEPT "请输入检索条件" TO SR
ENDDO
```

函数TYPE产生SR中表达式的类型，在dBASEⅡ中产生的类型是C（字符型）、N（数字型）、L（逻辑型）三种，只有输入表达式的类型为逻辑型（如民族="汉".AND.职称="高工"等）。

才不重新输入。本程序段适应于严格要求表达式为逻辑型的情况。

8.3.3 宏替换的技巧

在 dBASE 中的宏替换函数用途很广，灵活、巧妙地使用，能增加程序的灵活性、通用性、减少程序量。现将其用途归纳如下：

1. 变量代换

在管理程序中，程序长而繁，利有&函数可使程序精练。

例如：对 BB.DBF 的 B1～B9 中的 9 个字段进行计算的程序如下：

```
SELECT PRIMARY
USE B1
SELECT SECONDARY
USE B2
STORE O TO N
SELECT PRIMARY
DO WHILE N< =9
N=N+1
STORE STR(N,1) TO X
REPLACE ALL TO TAL&X WITH S.SUB&
X * S.XS&X
ENDDO
```

这比用 9 个 REPLACE 语句精简得多

用宏替换替换内存变量的简单例子：

```
X=123.4
Y='X'
Z=10+&Y          (Z 的值 133.4)
```

在第三条语句中，&Y 替换 X 的值，即 Z=10+123.4

2. 替换字段的内容

在程序中，有时要根据键入值确定显示当前记录某字段的内容。如：已知数据库 EXA.DBF 的结构由 A1、A2、A3、A4 共 4 个字段构成，则可用如下程序段显示该库当前记录某字段的内容：

ACCEPT ‘字段编号：’ TO I

X＝‘A’+I

? &X　　　（显示第 I（1<I<4）个字段的内容）

3. 替换字段列表

替换字段列表的方法是，将字段表（字段名间用逗号“，”隔开）以字符串形式赋给一个内存变量，然后用该变量宏替换该字段列表。仍假定字段名为 A1、A2、A3、A4，其语句序列可以是：

X＝‘A1，A2，A3’

LSIT &　　　（列表）

BROWSE FIELDS &X　　　（编辑）

COPY FIELDS &X TO EX2　　　（生成一个新库）

4. 替换逻辑表达式

dBASE 中有许多语句（如 COPY，LOCATE，DELETE 等）可用一个逻辑表达式（FOR 子句）指定语句的处理范围。如：

USE EX2

ACCEPT ‘输入检索条件编号：’ TO I

DO CASE

CASE I＝‘1’

```
X = ‘A2 > = 10.AND.A3 < = 5’
CASE I = ‘2’
X = ‘A2 > 2.AND.A3 < = 10’
CASE I = ‘3’
X = ‘A3 < 5.AND.A4 > 0’
ENDCASE
LOCATE FOR &X
…
```

5. 替换一条语句

当一条语句在多个地方出现时，用此种方法可缩减程序长度，减少输入量。如：

```
X = ‘DO WHILE .NOT.EOF
…
&X          (等价于语句：DO WHILE. NOT.EOF)
…
ENDDO
```

6. 替换一个字符序列

在字符序列中用宏替换插入内存变量（字符型）的内容：

X = ‘WANG’

Y = ‘ HELLO & X’　　（运行结果：Y = ‘HELLO WANG’）

若宏替换的内存变量名后还有字符，则用以园点‘·’表示内存变量名的结束。如

X = ‘+’

Y = ‘12.2&X.4’　　（运行结果：Y = ‘12.2+4’）

总之，宏替换函数的用途是非常广泛的，如果运用得

当，它会带来很大方便。

8.3.4 巧建屏幕格式

dBASE 具有建立格式文件供多处调用的特点。

1. 建立格式文件 FORM

```
·MODIFY COMMAND FORM.FMT
@ 1, 20 SAY'* * * * * * * * * * * * * * * * *'
@ 2, 20 SAY'*                               *'
@ 3, 20 SAY'* DBASE FORMAT                  *'
@ 4, 20 SAY'*                               *'
@ 5, 20 SAY'* * * * * * * * * * * * * * * * *'
∧W
```

从上面建立格式文件的过程可见，它的建立过程和命令文件的建立过程类同，区别有两点：第一点，格式文件的文件名后面要加上扩展名"·FMT"。第二点，格式文件的结束处不含"RETURN"命令。

2. 格式文件的调用

调用格式文件的两个命令是：

```
SET FORMAT TO 〈文件名〉
READ
```

当命令文件执行到第一个命令时，就把格式文件取出来，当第二命令即 READ 执行时，则执行格式文件中的"@"命令。

假定命令文件 FILE1.CMD 的内容如下：

```
SET SCREEN NO
SET FORMAT TO FORM
READ
```

```
STORE 1 TO NO
DO WHILE NO<200
STORE NO+1 TO NO
LOOP
ENDDO
ERASE
SET FORMAT TO FORM
READ
RETURN
```

在这个命令文件中，两次调用格式文件 FORM。

8.3.5 构造数组

dBASE 不具有数组功能，使用宏代换可模拟数组功能。

下面是一个数组运算的程序，设数组 A(2，3)，数组 B(3，4)，其值为

$$A=\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{pmatrix} \qquad B=\begin{pmatrix}1 & 3 & 5 & 7\\ 2 & 4 & 6 & 8\\ 3 & 5 & 7 & 9\end{pmatrix}$$

求 C=A＊B

```
STORE '1' TO I
DO WHILE &I< = 2
STORE '1' TO J
DO WHILE &J< = 3
INPUT A&I&J
STORE STR(&J+1,1) TO J
ENDDO
```

```
STORE STR(&I+1,1) TO I
ENDDO
STORE '1' TO I
DO WHILE &I<=3
STORE '1' TO J
DO WHILE &J<=4
INPUT B&I&J
STORE STR(&J+1,1) TO J
ENDDO
STORE STR(&I+1,1) TO I
ENDDO
STORE '1' TO I
DO WHILE &I<=2
STOR '1' TO K
DO WHILE &K<=4
STORE 0 TO C&I&K
STORE '1' TO J
DO WHILE &J<=3
STORE C&I&K+A&I&J * B&J&K TO C&I&K
STORE STR(&J+1,1) TO J
ENDDO
STORE STR(&K+1,1) TO K
ENDDO
STORE STR(&I+1,1)TO I
ENDDO
LIST MEMO
QUIT
```

本程序，第一个二重循环，构造数组 A 及其数据；第二个二重循环，构造数组 B 及其数据。这些数据是由键盘逐一输入的。第三个三重循环求出数组 C 各元素的值。然后用 LIST MEMO 将数组 C 显示出来。

8.3.6 设计口令

为了保密，或防止一些数据被误操作所破坏，在修改数据或程序启动之前设计口令。

1. 把口令字符串直接安排在程序中

```
ERASE
SET TALK OFF
STORE " " TO KL
? "请输入口令!"
SET CONSOLE OFF
ACCEPT TO KL
SET CONSOLE ON
IF KL<>"BMKL".AND.KL<>"bmkl"
? '口令有误'
QUIT
ENDIF
```

该程序段很简单，一旦输入的口令不是 BMKL，也不是 bmkl，程序控制退出 dBASE。

2. 把口令放在内存变量文件中

程序如下:

①建立和修改口令的程序段

```
ACCEPT "请输入口令" TO LOCK
STORE !(LOCK) TO LOCK
```

```
SAVE TO XX.MEM
RETURN
```

②审核口令的程序段

```
SET TALK OFF
RESTOR FROM XX.MEN
ERASE
?"请输入口令!"
SET CONSOLE OFF
ACCEPT TO MKL
SET CONSOLE ON
IF LOCK<>!(MKL)
? "口令有误"
CANCEL
ENDIF
```

本程序审核口令有误，中止程序运行；如发现失密，可随时运行建立口令的程序段，建立新口令。

3. 允许输入一定次数的有误口令

假定仍由 2 中的程序段建立口令，审核口令的程序段可设计如下：

```
RESTORE FROM XX.MEM
SET TALK OFF
STORE 0 TO M0
DO WHILE M0<5
REMARK    请键入口令!
SET CONSOLE OFF
ACCEPT TO M1
SET CONSOLE ON
```

```
IF ！ (M1)=LOCK
STORE 5 TO M0
ELSE
?“口令有误”
STORE M0+1 TO M0
ENDIF
ENDDO
IF ！ (M1)<>LOCK
?“全是有误口令，仃止运行”
CANCEL
ENDIF
...
```

本程序段允许使用者输入5次口令，若其中有一次口令正确，则往下执行程序；若5次口令都不正确，则停止程序运行。

以上介绍的方法可以在不同的程序中使用，可以根据不同密级设置多重关口。还可以采用多个口令调用同一数据库的方法，各用各的口令，把使用次数记录下来，以备检查。还可以必须两人以上，各用各的口令，口令都对上，才能进入。还可以将要保密的资料经过某种变换后保存，把变换方法的程序存入软盘，在使用数据前，必须要软盘上的有关程序变换转来才能使用。总之掌握基本方法后，就可以根据需要进行设计。

8.3.7 建立专用词库

在应用程序中，有时需要多处和多次使用某些词和短语，把这些词和短语组织起来，建成专门的词库。可以简化

输入输出，减少用户运行程序时因输入不正确造成的出错机会，简化程序操作。

假定某单位有 40 多个下属单位，为方便用户操作和减少各原始库的容量，设计一个存有各下属单位代码和单位名称的词库。在用户指定单位的地方，只键入有关的代码，程序自动取出单位名供使用。在数据库 DWK.DBF 中第 1 字段 DWH1 为前 8 个下属单位的单位码，第 2 字段 DWM1 为这 8 个下属单位的单位名；第 3 字段 DWH2 为 9～16 个下属单位的单位码，第 4 字段 DWM2 为这 8 个单位的单位名。依此类推，直到第 11 字段是 DWH6 最后 8 个下属单位的单位码，第 12 字段 DWM6 为最后 8 个下属单位的单位名。设计 8 个记录为的是不翻屏幕。

其程序段如下:

```
* 程序名: DW.CMD，将单位名存入 DWM 中
USE DWK
ERASE
LIST NEXT 8 OFF
INPUT "请输入单位码" TO DM
DO CASE
CASE DM<9
GO DM
DWM=DWM1
CASE DM>8.AND.DM<17
GO DM-8
DWM=DWM2
CASE DM>16.AND.DM<25
GO DM-16
```

```
DWM=DWM3
CASE DM>24.AND.DM<33
GO DM-24
DWM=DWM4
CASE DM>32.AND.DM<41
GO DM-32
DWM=DWM5
CASE DM>40.AND.DM<49
GO DM-40
DWM=DWM6
ENDCASE
RETURN
```

可见，用户在屏幕的提示下输入代码比输入汉字的单位名方便得多。

8.3.8 巧用 TOTAL 命令

dBASE 中的 TOTAL 命令功能很强，适用于许多分类求和的情况。

1. 按多关键字分类求和

由于 TOTAL 命令只能按一个关键字段进行分类求和，不能满足某些应用的需要，但只要对数据库结构改动一下就可按多关键字段分类求和。方法是把有关的字段按其主次进行字符串连接，构成一个新字段，再以这个新字段作为索引和分类求和的关键字段。如：

```
REPLACE ALL DD WITH AA+BB+CC
```

以 DD 字段作为索引和分类求和的关键字段（这里的 AA、BB 和 CC 是希望作为关键字段分类求和的三个字

段，DD 为供索引和分类求和的新字段)。这样采用：

TOTAL ON DD TO EE，

EE 就是所需要的结果。

2. 用 TOTAL 命令代替 JION 命令

由于 JION 命令时间开销大，有时失去了实用性，巧用 TOTAL 命令可在短时间内达到目的。

假定 A 库有字段工号，基本工资，实发工资，B 库有字段姓名，工号。用 TOTAL 命令产生一个 C 库，其字段为姓名，工号，基本工资，实发工资。

实现的步骤是：

第一步。将 A 库用 MODI STRU 增加姓名字段，宽度与 B 库姓名字段相同。

第二步。将 B 库用 MODI STRU 增加与 A 库相比没有的字段，基本工资，实发工资，宽度与 A 库的对应字段相同。

此时，字段内容为空，B 库除姓名和工号字段外，其他字段为空。

第三步。用 APPEND FROM 命令将 A 库添加到 B 库中。

第四步。将 B 库以工号关键字段（用命令 SORT ON 工号 TO D）排序产生 D 库。

第五步．对 D 库以工号为关键字段分类求和(TOTAL ON 工号 TO C FIELDS 基本工资，实发工资)。用 FIELDS 指定求和字段，为的是提高速度。C 库是一个字段为姓名，工号，基本工资，实发工资的新库。相当于 JOIN 命令的效果。

8.3.9 美化屏幕显示

所谓美化屏幕就是如何利用屏幕，把屏幕的显示搞得精美好看。随着计算机应用的深入广泛，人们对美化屏幕越来越感兴趣。dBASE 在美化屏幕方面的功能比 BASIC 要差些，但仍可以作些努力，使屏幕生动活泼。下面给出的是一个软件的序幕程序段:

```
...
LS=1
DO WHILE LS<=100
DO CASE
CASE LS<=40
@ 0, 40-LS SAY "%"
@ 0, 39+LS SAY "%"
CASE LS>40.AND.LS<=48
@ LS-40, 0 SAY "%%%%"
@ LS-40, 76 SAY "%%%%"
CASE LS>48.AND.LS<=88
@ 8, LS-49 SAY "%"
@ 8, 128-LS SAY "%"
ENDCASE
LS=LS+1
ENDDO
@ 2, 24 SAY "中"
DO YS
@ 2, 28 SAY "药"
DO YS
@ 2, 32 SAY "识"
```

```
DO YS
@ 2, 36 SAY "别"
DO YS
@ 2, 40 SAY "专"
DO YS
@ 2, 44 SAY "家"
DO YS
@ 2, 48 SAY "系"
DO YS
@ 2, 52 SAY "统"
DO YS
@ 4, 30 SAY "作者
DO YS
@ 4, 37 SAY
"××××××××××××××"
DO YS
@ 6, 27 SAY "浙江科学技术出版社"
DO YS
DO YS
…
```

延时程序段 YS:

```
H = 20
DO WHILE H > 0
H = H-1
ENDDO
RETURN
```

执行本程序段时，首先动态的在屏幕构成一个边框。先

从上边线的中间向两边，接着是左右边线同时进行，最后是从下边线的两头向中间布上百分符“%”，上、下为一排百分符，两边各为四列百分符构成的竖线。

然后是在第二行逐个显示：中药识别专家系统，接着在第四行显示：作者，最后在第六行显示：浙江科学技术出版社。它们之间的时间间隔是调用延时程序 YS 来实现的。可将百分符改为其它喜欢的字符，修改语句 H=20 可改变延时的长短。

8.3.10 如何调试 cdBASE 程序

程序写好之后，须进行测试和纠错。

1. **可能的问题**

通常有这样几类错误：①命令拼写错，②命令和表达式之间无空格，③句法错，即命令使用不正确，④命令不完整，⑤运行错。

克服拼写错和命令与表达式之间无空格的方法是，打印源程序，仔细校对。错误③或④比较难以确认，提供参考的方法有：

①检查程序中内存变量名、文件名和字段名的拼写错。

②注意不匹配的数据类型。例如：用同一个@命令显示数值和字符型字段，对此用转换函数纠正；对非数值变量执行数值（函数）运算，对此将变量用数值转换予以纠正。

③仔细检查语句中的顺序是否错。

④检查语句中的选择项是否有错。

⑤命令一般可简化为 4 个字符。如：ENDCASE 和 ENDC 都是可以的，而 ENDCA 就是错误的。

运行期间的错误多是逻辑方面的，其原因不是没按

dBASE 要求写，就是逻辑上与 dBASE 不一致。检查是否误用了逻辑运算符，.AND.和.OR.，特别是当它们在复杂的逻辑表达式中有时就容易出错。

2. 调试的步骤

模块化的程序设计方法，可在编写这些模块程序时，同时进行调试。从整体上说，调试程序可采用如下方法步骤:

①写出每一模块，并使其文件化，一旦写完就加以测试。

②使用带有其它技术的内部调试特性来调试程序模块。

③当测试完一个模块并运行正确时，继续下一模块。

④当完成相互配合的模块后，将它们联系起来，作为一个组合程序进行测试。

⑤逐步增加新的测试好的模块，建立更大的组合程序，进行相应的测试。

⑥整个项目全部程序装配完后，同样进行测试。

⑦把程序交给非作者的其他人测试，称为 α 测试。

⑧交给用户测试，称 β 测试。

⑨交互使用。使用中遇到软件问题，还要进行维护修改。

装配时候和设计阶段自顶向下相反，自底向上进行。

在 α 和 β 测试中，发现的错误最好让程序作者修改，因为作者比较清楚程序内部结构，比较好避免修改中增加新的错误。在动手修改之前，应考虑到程序和数据备份，以防调试的程序不能运行。有时，调试一个程序模块最有效的方法是，重写整个程序模块或有关部分。

3. 有关的调试命令

下面列出一些与调试有关的命令，可使用这些命令，调

试dBASE程序。

①LIST STRUCTURE

作用：显示数据库的结构。

②LIST MEMORY

作用：显示内存变量的内容。

③SET TALK ON

作用：显示执行各条命令的结果。

④SET STEP ON

作用：使dBASE执行一条语句就停一下。

⑤SET ECHO ON

作用：使执行的语句在屏幕上显示。

⑥SET ALTERNATE TO〔〈文件名〉〕

SET ALTERNATE ON

作用：执行这两条命令后，把屏幕上或打印机上出现的信息用指定的文件名存在磁盘上。

⑦SET PRINT ON

作用：接通打印机。

4. 调试方法

①单步执行

在程序执行前，执行如下的命令：

SET STEP ON

SET ECHO ON

SET TALK ON　　(或SET DEBUG ON)

再运行程序时，一条一条地执行，能观察执行顺序及结果是否正确。

②设“断点”调试

分析出有可能出错的位置设“断点”（用WAIT命令）

及显示的提示信息，说明位置前面执行正常，请再继续，……等。

例如：

……

?"第 n 段程序执行正常，请按任一键继续!"

WAIT

……

?"第 n+j 段程序执行正常，请按任一键继续!"

WAIT

……

③用 ESC 键中断运行，查看"现场"。

在执行中，若到某一个"断点"（WAIT）处，发现有错，或不知是否正确？想要查看"现场"，按 ESC 键中止程序运行，用显示或打印命令，检查内存变量、程序循环次数、数据库中的数据、记录号的值等等。

④用编译 dBASE 编译查错

有 dBASE 编译软件时，可以用编译 dBASE 编译 dBASE 程序，查出错误。编译查错一般都可以查出多个或全部错误。

习题八

1. 结合自己学习或工作实际中的一、二个实例理解程序及程序设计的概念。

2. 设计一个口令程序。

3. 设计一个美化屏幕的程序。

第九章　CP／M 操作系统

9.1　CP／M 操作系统概述

操作系统是协调和管理计算机系统资源（硬、软件设备），为用户提供方便的程序系统。

CP／M（Control Program／Monitor 的缩写）是为 Z80 及 8080 微处理机而设计的操作系统程序。它的一个特色是适合在任何以 8080 或 Z80 为中央处理机，主存容量在 16K 以上，配有 1～4 个与 IBM 兼容的磁盘驱动器的系统上运行使用。

cdBASEⅡ（也称汉字 dBASEⅡ，下同）是在 CP／M 操作系统下运行的，要使用 cdBASEⅡ，需要对 CP／M 操作系统有所了解。

9.1.1　Z80 插件

由于 cdBASEⅡ需要 CP／M 操作系统的支持，而 CP／M 操作系统又必须用到 Z80 或 8080CPU。因此，在采用 6502 为 CPU 的中华学习机上运行 cdBASEⅡ，首先是要使中华学习机具备能运行 CP／M 操作系统的 Z80 或 8080CPU 的硬件环境。为适应广大用户的需要，有关厂家

设计了一种 Z80 插件，其中含有一片 Z80 微处理器，把这种插件插入中华学习机的扩展槽，即可运行 CP／M 操作系统及 cdBASEⅡ。这种 Z80 插件和 cdBASEⅡ软件本部可向读者优惠提供。

9.1.2 CP／M 操作系统的版本

早在 1973 年，Gary Kildall 首先研制出 CP／M 操作系统，几经扩充发展，成为 8 位微机中应用非常广泛的操作系统，为了适应各种机型的需要，发展了多种版本。

在 CP／M 版本的表示法中，用小数点左面的数字代表整体的型号区别，小数点右边的数字代表同一型号的修订版，小数点右边第二个数字代表版本上的细微差别或与具体机型有关的版本。

CP／M－80 版本能用于 8080、8085 和 Z80 的 CPU，CP／M－86 版本适用于 8086 和 8088 型 CPU。

CP／M－80 版本主要有:

1.3　最原始的 CP／M－80 版本;

1.4　版本 1 经修订后的版本;

2.0　版本 2 的原始型;

2.1　版本 2 经过修改后的版本;

2.2　版本 2 的修订版。

以上是按 Digital Research 公司对于 CP／M 的修改版号分类的，有些厂商的自行修改，在右边第二位小数上表示出来。

由于各种微机系统配置不同，在使用 CP／M 操作系统时，系统文件的组成略有差异，可能会有不同格式的系统磁盘。中华学习机的 CP／M 操作系统磁盘上只保留了

CP／M 的主要模块和 cdBASE 软件。这是为了用户在软盘上有更大的可用空间，特地将操作系统的一些不常用的文件，如 FORMAT 及其它一些外部文件分离开来。

9.1.3 CP／M-80 操作系统盘

在 CP／M 系统盘上有许多处理程序，供用户需要时使用，现将主要程序的作用说明如下：

1. ASM CP／M8080 汇编程序。用于处理 8080 汇编语言程序。

2. CONFIGIO 建立适合用户需要的 CP／M 操作环境。

它有四个主要功能：①为外来终端机沟通输入或输出(I／O)；②重新定义键盘字符；③装入用户的 I／O 软件；④读入或写入 I／O 结构块。

3. COPY 用于复制 CP／M 系统磁盘，即把 CP／M 系统程序复制到新的格式化好的空白磁盘上，产生一个新的 CP／M 系统盘。

4. DOWNLOAD 在两台都使用 CP／M 操作系统的微机之间，可用本实用程序通过 R-232 串行数据传输接口进行通讯。

5. DUMP 将磁盘文件内容以 16 进制数字形式显示。

6. ED 用于编辑文本文件的文本编辑程序。

7. FORMAT 用于格式（初始）化磁盘。在用新盘存入信息之前必须进行格式化。

8. LOAD 将扩展名为.HEX 的磁盘文件转换为机器可执行（扩展名为.COM）的文件；还能将汇编程序编译输出转换成可执行的机器代码。

9. MBASIC 是 Microsoft BASIC 解释程序。提供低分辨绘图功能、声音和游戏控制。无高分辨率绘图功能。

10. PIP 用于磁盘间的文件复制；或将文件移到终端或打印机，实行显式或打印。还可复制及加长磁盘文件。

11. STAT 提供磁盘状态信息:①磁盘容量;②文件大小;③文件指示器;④修改文件指示器及配件安排等。

12. SUBUIT 用于通过磁盘文件中的命令和程序执行操作，达到系统自动处理的目的。

13. XSUB 与 SUBMIT 共用，可在执行程序时，随时从磁盘文件中输入字符。

14. CPM56 当 44K 系统上有语言卡时，利用 CPM56 把系统变为 56KCP／M 操作系统（44K 再加上语言卡 12K）。56K 系统磁盘指的是具有 RAM（内存）为 64K 的微机上用的磁盘。可见 CPM56 不能用于 48KRAM 的系统中。

在中华学习机上运行的是 56KCP／M 操作系统盘，不需使用 CPM56 程序。

15. GBASIC 除具有 MBASIC 功能外，还能提供高分辨率绘图功能。

16. RW13 使 16 扇区 CP／M 存取 13 扇区 CP／M 磁盘上的文件；与 PIP 合用，在有两台以上磁盘机的系统上，可将 13 扇区磁盘上的文件转移到 16 扇区的磁盘上。

17. APDOS 用于将 APPLEDOS 盘上的文本文件及二进制文件数据传送到 CP／M 磁盘上。

9.1.4 建立 56KCP／M-80 系统

用于 48KRAM 的 44KCP／M-80 操作系统，可以用

下述的方法修改为用于 64KRAM 的 56KCP／M-80 系统盘。方法是:

①为防止意外的损坏 44K 系统，先复制一张 44K 系统盘（复制方法参见本章第 9.4.3 复制磁盘命令 COPY），下面的工作在复制盘上做。

②将复制的 44KCP／M 系统盘插入 A:驱动器，开机显示 A>。

③键入命令:

A<CPM56A:↓

微机自动建成 56K 的 CP／M 系统盘。

因 CEC-I 中华学习机是具有 64KRAM 的系统，故使用的已经是 56KCP／M 操作系统盘。

9.1.5 Z80 卡的安装

对于 CEC-I 中华学习机扩充插槽插入 Z80 卡分两种情形。一种情形是没采用 50 线转接板把槽口抬高；一种情形是选用一个 50 线转接板把槽口抬高。对于前者需要事先卸下学习机的外壳。

插入 Z80 插件的一般步骤是:

①关掉主机电源。

②双手拿住插件顶端垂直插入插槽，注意不要插错方向；不要玷污插件的插脚，确保插件与插槽接触可靠。

③检查无误，再开机。

9.1.6 CP／M 操作系统的文件命名规则

CP／M 系统的文件命令规则，不同于中华学习机汉字 DOS3.3 磁盘操作系统。其主要规则有:

①每个 CP／M 文件名由两部分组成，即文件名和扩展名，中间用小数点符号隔开。还可在文件名前加上驱动器名，以说明这个文件是在那个驱动器的磁盘上，若不指定驱动器名，系统就认为文件是在“当前驱动器”上。文件名必须用大写字母，也可用数字，最多只能用 8 个字符，扩展名最多只能用 3 个字符，也可以省略扩展名。下面列出的是一些合法的文件名例子。

FILE1·BAS　　文件名 FILE1，扩展名 BAS

80860·COM　　文件名 80860，扩展名 COM

AB-12·ASM　　文件名 AB-12，扩展名 ASM

下面是常用的扩展名：

ASM　　表示用汇编语言写成的文件

ASC　　ASCII　字符文件

BAS　　BASIC　语言文件

BAK　　后备文件

COM　　可立即执行的文件

DAT　　数据文件

FOR　　FORTRAN　语言文件

PRN　　打印或列表文件

LIB　　库文件

HEX　　十六进制格式文件

②文件中不能含有如下的字符：

〈〉　·，；：　＝？＊〔〕　（）　／或〈TAB〉

③文件名中不能使用控制字符或其它无法在屏幕上显示的字符。

9.1.7　CP／M 操作系统的文件访问方式

CP/M 系统具有单个或成组两种访问文件的方式。

1. 单个访问就是在命令中直接用所要访问的某一文件的名字，例如：运行驱动器 A:上的 DBS.COM 文件，就在 A>提示符下，直接敲入命令：

A>DBS↓

其中系统规定可执行文件在运行命令中可以省掉扩展名。又如：在屏幕上显示 DBS.COM 文件，就在 A>提示符下，直接用显示命令及文件名 DBS.COM，即：

A>DIR DBS.COM↓

2. 成组访问磁盘文件，可以提高磁盘操作效率，又可以分为两种方法：

①用星号"*"代表文件名或扩展名中一串字符。

. 代表磁盘上所有文件。

*.BAS 代表磁盘上所有扩展名为 BAS 的文件。

FILE.* 代表所有名字为 FILE 的文件。

F*.* 代表所有第一个字符为 F 的文件。

②用问号"?"代表文件名或扩展名中的一个字符。例如：

F? B.FOR 代表除？处（文件名的第二个字符位置）字符不同以外的所有同名文件。

YFSB.??? 表示所有名字为 YFSB 的文件。

???????? .??? 与*.*作用相同。

9.2 中华学习机 CP/M 操作系统的启动

对于按 9.1.5 的步骤插好 Z80 卡的中华学习机，可以按

下述步骤来启动 CP／M 操作系统。

9.2.1 冷启动

冷启动（也称开机引导）的步骤是:

1. 打开显示器或电视机的电源开关。

2. 把中华学习机 CP／M 操作系统盘插入驱动器 A:。

3. 打开主机电源，驱动器灯亮，CP／M 系统自动进入主机，屏幕显示:

APPLE Ⅱ CP／M

56K VER 2.20B

(C) 1980 MICROSOFT

A >

至此，完成了 CP／M 操作系统的启动。A > 是 CP／M 操作系统的提示符。

9.2.2 热启动

操作系统的热启动（又称热引导），其步骤是:

1. 打开显示器或电视机的电源开关。

2. 打开主机电源，屏幕上出现 BASIC 语言提示符“]”。

3. 打入命令:

A > PR#6↓

即可见到驱动器灯亮，CP／M 系统自动进入主机，屏幕显示如 9.2.1 冷启动。

CP／M 操作系统启动之后，可接受并执行有关命令。

9.2.3 当前驱动器

当屏幕上出现提示符“A > ”时，A 表示当前驱动器是 A:。如果在命令中不指定驱动器，系统约定为“当前驱动器”。当前驱动器可以更换，更换的方法是键入驱动器名及冒号“:”。

例如:

A > B:↓

其中 A > 是屏幕原有的提示，B:↓是使用者键入的。屏幕即显示:

B >

表明系统已经把当前驱动器由 A 更换成 B。

对于出厂未扩充的 CEC–I 学习机一般只有一个驱动器 A:，不宜做更换操作。

9.3 CP／M 内部命令及使用

CP／M 的内部命令也称直接命令，是在提示符 A > 之下直接执行的命令。一般来说它们是常驻内存的。

9.3.1 内部命令的用法

1. 列文件目录 DIR

格式: DIR↓

作用: 显示磁盘文件目录。

如果显示当前驱动器中磁盘上所有文件目录，则键入: DIR↓。如要显示其它驱动器上的文件名，需要加上驱动器名，如:

A > DIR B:↓

显示驱动器 B:上的全部磁盘文件目录。

A>DIR *.BAS↓

显示当前驱动器中的所有扩展名为 BAS 的文件目录。

A>DIR F*.*↓

显示当前驱动器中的所有文件名第一个字符是 F 的文件目录。

2. 删除文件 ERA

软盘上没有保存价值的文件，可用删除命令删除掉，删除文件省下的容量，可用于保存新文件。

格式：ERA〈文件名〉↓

作用：删除磁盘中指定的文件。

其中文件名是必须的。例如：

ERA DEF.*↓

删除当前驱动器中文件名为 DEF 的所有文件。

ERA B:FILE.COM↓

删除驱动器 B:中名为 FILE.COM 的文件。本命令指定驱动器 B:，不能用于只有单驱动器的情形。

ERA*.*↓

删除当前驱动器中所有文件，此时显示提问：

ALL (Y/N)?

如答 Y，则执行删除；回答 N，则不执行删除。

3. 重新命名文件名 REN

格式：REN〈新文件名〉=〈老文件名〉↓

作用：用新文件名替换老文件名。

文件名之前也可指定驱动器，例如：

A>REN FILE.NEW=FILE.OLD↓

将当前驱动器中原名叫 FILE.OLD 的文件改名为

FILE.NEW。

注意：不能颠倒新老文件名的位置。命令中新文件名应是在相应驱动器磁盘上原来不存在的文件名。

4. 显示文件 *TYPE*

格式：TYPE〈文件名〉↓

作用：显示磁盘上的 ASCII 码文件。

例如：

A>TYPE ABC.BAS↓

显示文件 ABC.BAS 的内容。

A>TYPE FILE.ASM↓

显示文件 FILE.ASM 内容。

说明：本命名中的文件名前面也可以指定驱动器名。

5. 保存文件 *SAVE*

格式：SAVE n〈文件名〉↓

作用：将用户区中第 n 页（256 字节为一页）内容用指定的文件名保存到磁盘上。在 CP/M 分布式系统中，TPA 从 100H 开始（存储器的第二页），因此，如果用户程序区占用 100H～2FFH，则 SAVE 命令必须指定存储器的 2 页，接下去装入和执行二进制文件。以下是 SAVE 的例子：

A>SAVE 3 XY.COM↓

将 100H～3FFH 的内容用 XY.COM 为文件名保存到磁盘上。

A>SAVE 39 FILE↓

将 100H～27FFH 的内容用 FILE 为文件名保存到磁盘上，其中 39 是十进制，27 为十六进制，即：$(27)_{16}=(39)_{10}$。

说明：命令中的 n 采用十进制；反复使用 SAVE 命令不改变存储器内容；文件名前可以指定驱动器名。

9.3.2 内部命令出错信息

在执行 CP／M 内部命令时，可能出现如下的错误：

1. NO FILE, NOT FOUND 或 FILE NOT FOUND

表示磁盘上没有命令所要求的文件。

2. BDOS ERR ON X:

X:代表驱动器名字，如 A:。表示 CP／M 没有找到命令中指定的磁盘或驱动器，说明磁盘装置有问题，也许是没有放好磁盘或磁盘没有做过格式化等。如：

BDOS ERR ON X: DISK R／O

表示磁盘装置有错。原因是：

①换了磁盘后没有执行引导。

②磁盘有写保护。

这时按压任意键，则会热引导，回到 A>。

BDOS ERR ON X: SELECT

不存在指定的驱动器。此时若键入任一字符，则热引导回到 A>。

BDOS ERR ON X: BAD SECTOR

表示磁盘装置有问题。键入 CTRL-C 则热引导。

BDOS ERR ON X: FILE R／O

表示指定的磁盘文件原先用 STAT 程序指定为只读文件，不能执行写入命令。此时键入任一字符，则执行热引导，回到 A>。

3. FILE EXISTS

在执行 REN 命令时，磁盘中已存在与指定新文件名同

名的文件。

9.4 CP/M 外部命令及使用

CP/M 外部命令的处理程序存储在外存磁盘上，执行时需要首先启动操作系统。如果这些命令不在 CP/M 系统盘上，而在另外的磁盘上，则需要在启动操作系统之后，取出 CP/M 系统盘，插入具有外部命令文件的磁盘。键盘输入命令的方法与输入内部命令一样。外部命令可以指定驱动器名。

9.4.1 复制文件命令 PIP

格式：PIP〈目标文件名〉=源文件$_{-1}$〔，源文件$_{-2}$，…，源文件$_{-n}$〕

作用：复制文件，连接产生新文件，输出文件。

PIP 是 Pcriphcral Interchange Program（外部交换程序）的缩写。PIP 命令有两种使用方法：

1. PIP↓

响应本命令的是在屏幕上显示 PIP 的提示符（星号）“*”，等待用户键入命令行，用户可以反复地输入命令行，直到在一行的开始键入回车为止。例如：

```
A>PIP↓
*FILE.BAK=FILE.TXT↓
*A:=B:AAA.BAS↓
*↓
A>
```

这种方式下执行完了一条 PIP 命令，又回到星号“*”

提示符，再等待键入一条命令。所以这种方式适用于多次执行 PIP 命令。若希望退出这种状态返回到操作系统。则使用 CTRL−C 或回车键回到 A＞。

2. PIP〈命令行〉↓

这是直接执行 PIP 命令方式，适用于一次性作业，作完之后自动回到操作系统提示符 A＞。

例如:

A＞PIP FILE.BAK＝FILE.TXT↓

A＞

上述两种使用法的具体用途是:

1. 把文件复制到磁盘上。

①PIP X:〈新文件名〉＝Y:〈老文件名〉〔参数〕↓

把驱动器 Y:磁盘上的老文件复制到驱动器 X:的磁盘上，且换成新的文件名。例如:

A＞PIP EFG.DBF＝ABCD.DBF↓

作用是把 ABCD.DBF 文件复制一个名为 EFG.DBF 的文件。即磁盘上增加了一个 EFG.DBF 文件，其内容与 ABCD.DBF 相同。

A＞PIP B:FILE2.BAS＝A:FILE1.BAS↓

将驱动器 A:磁盘中的 FILE1.BAS 文件复制到驱动器 B:磁盘中，名字为 FILE2.BAS。

A＞PIP B:FILE2.BAS＝A:FILE1.BAS〔V〕↓

把 A:中 FILE1.BAS 文件复制到 B:中磁盘上，名字改为 FILE2.BAS，参数 V 表示复制好后，自动核对新老文件是否一致，核对结果正确无误，便出现 CP／M 系统提示符。这种办法用于检查复制是否正确。

②成组复制文件

PIP 命令也可以用星号“*”成组复制文件。例如：

A > PIP　B: = A: * .DBF〔V〕↓

显示：

COPYING—

A.DBF

AB.DBF

DATA.DBF

A >

其执行结果是把驱动器 A:中磁盘上所有扩展名为 DBF 的文件复制到 B:中，文件名不变。

③几个文件连接后复制，建立一个新文件。

例如：

A > PIP　AAA.BAS = BBB.BAS,CCC.BAS,DDD.BAS ↓

本命令的执行结果是：将当前驱动器中软盘上的 BBB.BAS,CCC.BAS 和 DDD.BAS 三个文件连接起来，复制建立新文件 AAA.BAS。

注意：连接时要求各个文件均为 ASCII 文件，命令行的长度不超过 255 个字符。

2. 沟通磁盘文件与其他外部设备关系，或沟通外部设备间关系。

①PIP 命令访问各种设备的约定代码如下：

TTY	控制台，输入机或行打机
CRT	控制台显示器
UC1	控制台
PRT	纸带输入机
UR1	输入机

UR2 输入机

PTP 穿孔机

UP1 穿孔机

UP2 穿孔机

LPT 列表机

CON 控制台

RDR 输入机

PUN 穿孔机

LST 列表机

②实例

A > PIP LST: = A:AABB.BAS ↓

作用是将驱动器 A:中磁盘上文件 AABB.BAS 从打印机输出，按任一键则终止输出。

A > PIP CRT: = A:AABB.BAS ↓

将文件 AABB.BAS 从控制台显示器显示输出。

3. PIP 命令参数的使用

在 PIP 命令中，用户可以指出一个或一组参数，并用方括号把这些参数括起来，用“0”或较多的空格分开。括号中的一组参数要紧跟在所起作用的文件名或设备名之后。除“S”和“Q”两参数外，参数后面还可跟一个十进制整数。PIP 命令的各种参数及其功能如下:

(1)B 成组传送方式。将源文件存入内存缓冲区，直到键入 CTRL–S 后建立并传送到目的设备，缓冲区能缓存的数量依主系统的内存容量。若缓冲区溢出，PIP 就显示出错信息。

(2)Dn 删去每行中第 n 个字符以后的所有字符，构成目标文件。

(3)E　屏幕上显示 PIP 的操作过程。

(4)F　删去文件中的“格式馈送”字符，它与 P 合用可以插入新的“格式馈送”字符。

(5)Gn　将代号为 n 的原用户区复制到当前用户区，n 为十进制数。

(6)H　十六进制数据传送。复制时审查是否正确的 Intel 十六进制文件格式。删去十六进制记录之间不必要的字符。若发现错误，则提示用户改正。

(7)I　除对“:00”的数据不作转换外，作用与 H 相同。

(8)L　将大写字母转为小写字母。

(9)N　给目的文件的每一行前面加一个行号，起始值为 1，增值也为 1。行号前零“0”省去，行号后面跟随一个冒号“:”。若参数为 N_2 则不省行号的前零“0”，并在行号后跟随扩展列表字符 tab（CTRL–I），若设定 T 参数则有扩展的 tab 作用。

(10)O　目标文件转换，CP／M 的文件结束符不起作用。

(11)Pn　在 n 行之后插入换页信息。默认值 n＝60。

(12)$Q_{s\uparrow z}$　复制时遇到字符串 S 则终止复制工作。（字符串 S 是以 Ctrl–Z 结束）。

(13)R　读系统文件。R 使有系统属性的文件在 PIP 中也能复制，否则不能复制系统文件。

(14)$S_{s\uparrow z}$　复制中忽略字符串 S 以前的内容，配合参数 Q 实现摘录文件的某一段。开始和终止复制的字符串均被复制。注意：若使用 PIP 命令的第二种形式，即有命令行的 PIP 命令，跟随±S 及 Q 参数后的字符串 S，自动转换成大写字母。若不采用有命令行的 PIP 命令，则不自动转换

字符串 S。

(15)Tn　在目的文件的每行第 n 个字符后插入扩展列表字符 tab（CTRL–I 字符）。

(16)U　复制中把小写字母转换成大写字母。

(17)V　复制后自动检验复制的内容是否有错。要求目的文件为磁盘文件。

(18)W　不提示用户自动改写文件属性为 R／O 的文件，若无 W，则在 PIP 过程中，询问用户是否改写。

(19)Z　将每个 ASCII 字符的奇偶位置成“0”。

使用 PIP 命令参数的例子：

①A＞PIP　A:＝B:ABC.ASM〔V〕↓

将 B:盘中的文件 ABC.ASM 复制到当前驱动器 A:，并校验有否错误。

②A＞PIP　PRN:＝ABC.ASM〔P55〕↓

将文件 ABC.ASM 打印输出，每页 55 行。

③A＞PIP　AB.LIB＝CD.ASM〔S_{SUB}R1:↑Z9JMPL3↑Z〕

把文件 CD.ASM 中的字符串 SUBR1 直到 JMPL3 的一段复制成文件 AB.LIB。

9.4.2　磁盘格式化命令 FORMAT

格式：FORMAT〔驱动器名〕↓

作用：格式化磁盘

新磁盘必须格式化后才能使用，对于信息较乱的旧盘，对其有保留价值的文件复制后，也可用格式化命令 FORMAT 将其格式化。根据系统驱动器配置情况有两种格式化方法：

1. 用一个驱动器完成格式化

当系统只有一个驱动器时，其格式化步骤是:

①把 CP / M 系统磁盘插入驱动器，启动 CP / M 操系统（详见 9.2 节中华学习机 CP / M 操作系统的启动）。注意：启动之后用 DIR 命令列文件目录，如当前 CP / M 操作系统盘上无 FORMAT.COM 文件，则取出该系统盘，插入具有 FORMAT.COM 文件的磁盘。

②打入命令:

A > FORMAT ↓

可见屏幕显示如下内容:

APPLE Ⅱ CP / M

16 SECTOR DISK FORMATTER

(C) 1980 MICROSOFT

INSERT DISK TO BE FORMATTED IN DRIVE A:

PRESS RETURN TO BEGIN

意思是要求把空白磁盘插入驱动器，再按回车键，格式化开始。若插入的是空白磁盘，则显示:

FORMATTING…

磁盘驱动器红灯亮，约 30 秒钟后，红灯熄灭，显示:

FORMAT COMPLETE

INSERT CP / M SYSTEM DISK IN DRIVE A:

PRESS RETURN

表示格式化工作已经作完，要求插入 CP / M 系统盘，按回车键回到 A >。

若键入格式化命令后系统发现用户插入的盘不是空白磁盘，存有信息，屏幕显示如下:

DISK IN DRIVE A: WILL BE ERASED

CONTINUE (Y/N)?

意思是提醒使用者：再执行就要删除磁盘上的信息，请再检查一下，若需要删除磁盘上的内容并格式化，则按Y键，否则按N键中止格式化。按Y键后磁盘显示同前述。

2. 用两个驱动器做格式化工作

在具有两个驱动器的机器上，格式化步骤如下：

①用CP/M系统盘启动系统，显示提示符A>。

②在驱动器A:插入具有FORMAT.COM文件的磁盘，键入命令：

A>FORMAT↓

屏幕显示：

APPLEⅡCP/M

16 SECTOR DISK FORMATTER

(C) 1980 MECROSOFT

FORMAT DISK IN WHICH DRIVE?

意思是已进入格式化程序，问要求格式化的空盘放在哪个驱动器？

③键入B:↓

则屏幕显示：

INSERT DISK TO BE FORMATTED IN DRIVE B:

PRESS RETURN TO BEGIN

意思是把空白磁盘插入B:并按回车键，若B盘是空白盘，屏幕显示：

FORMATTING…

驱动器红灯亮，完成格式化后显示：

FORMAT COMPLETE

FORMAT DISK IN WHICH DRIVE?

此时，若继续格式化其它磁盘，则按 Y 键格式另一张新磁盘，否则按回车键回到 A>；若发现插入驱动器 B:的是存有数据的磁盘，则屏幕显示:

DISK IN DRIVE B: WILL BE ERASED

CONTINUE (Y/N)?

要求确认是否格式化，若按 Y 键，格式化这张存有数据的磁盘，清除原数据；若按 N 键格式化中止。

CP/M-80 系统下经过格式化的磁盘可以存储信息，但并未置入 CP/M 系统，因此不能用来引导主机，这是与中华学习机 DOS3.3 系统格式化后的磁盘上就具有 DOS3.3 系统，可作为系统盘引导主机不同的。经 CP/M 系统格式化的磁盘，必须用 COPY 命令将 CP/M 系统复制到这种经格式化的磁盘，才能作为引导盘启动机器。

9.4.3 复制磁盘命令 COPY

格式: COPY〈目标文件〉=〈源文件〉↓

作用: 复制磁盘或文件

CP/M 操作系统复制磁盘的方法是：首先使用"FORMAT"命令格式化一片软盘（格式化操作参见9.4.2)。然后用 COPY 命令把原磁盘的内容复制到已格式化的备用盘上，这里介绍用 COPY 命令的两种复制方法:

*1. 复制 CP/M 系统盘上全部程序*将具有 *COPY* 命令程序的 *CP/M* 系统盘插入驱动器 *A:*，键入命令:

A>COPY A:=A:↓

屏幕显示:

APPLEⅡCP/M

16 SECTOR DISK COPY PROGRAM

(C) MICROSOFT 1980

INSERT MASTER DISK PRESS RETURN

意思是提示用户插入要复制的原盘，按回车键即开始复制。待按回车键后，驱动器灯亮几秒钟后，屏幕显示:

INSERT SLAVE DISK PRESS RETURN

提示用户取出原盘再插入格式化好的备用磁盘。再按回车键，几秒钟后，屏幕上显示:

INSERT MASTER DISK PRESS RETURN

提示用户取出备用盘插入原系统盘。如此重复几次，直到屏幕显示:

COPY COMPLETE

DO YOU WISH TO MAKE ANOTHER COPY?

(Y/N) PRESS RETURN

意思是复制工作完成了，问是否复制别的磁盘(Y/N)，按Y键表示重复上述步骤再复制一张CP/M系统磁盘；按N键则表示不再复制磁盘，系统控制返回到操作系统提示符A>。

2. 只复制CP/M操作系统

把含有COPY命令程序的系统盘插入驱动器A:，键入命令:

A>COPY　A:=A:/S↓

其中"/S"表示仅仅复制CP/M操作系统不复制系统盘上的其他程序。屏幕的显示是:

INSERT MASTER DISK PRESS RETURN

意思是插入系统磁盘，再按回车键。系统接受回车键

后，软盘驱动器 A:灯亮，读入盘中信息。屏幕将显示:

INSERT SLAVE DISK PRESS RETURN

提示用户（取出原盘）插入格式好的备份盘后，按回车键，系统接受回车键后，屏幕将显示:

INSERT CP／M SYSTEM DISK INTO DRIVEA PRESS RETURN

此时，按回车键就返回到 CP／M 系统。

9.4.4 显示设备状态命令 STAT

STAT 命令的使用功能，可归纳如下:

1. 统计显示磁盘文件及主机起动以来各驱动器的情况

格式: STAT↓

作用: 显示读／写属性及其可用空间。

例如:

A＞STAT↓

A:R／W，SPACE:12K

表示驱动器 A:中磁盘为读写磁盘，还有 12K 字节的存储空间供使用。

又如:

A＞STAT↓

A:R／0，SPACE:23K

B:R／W，SPACE:50K

表示驱动器 A:中磁盘为只读磁盘，还剩 23K 空字节；驱动器 B:中磁盘为读写磁盘，还有 50K 空字节。

2. 显示指定驱动器中磁盘的可用空间

格式: STAT X:↓

作用: 显示驱动器 X:中磁盘的剩余储存空间。

例如:

A>STAT　B:↓

BYTES REMAINING ON B: 123K

表示驱动器 B:中磁盘还有 123K 字节的存储空间。

3. 显示指定驱动器磁盘文件的大小和属性

格式: STAT　X:〈文件名〉↓

作用: 显示文件大小和属性。

例如:

A>STAT　FILE1.DBF↓

屏幕显示:

RECS　BYTES　EXT　ACC

10　5k　1　R/W　A:FILE1,DBF

BYTES REMAINING ON A: 150k

表明文件 FILE1.DBF 有 10 个记录，长度 5K 字节，占一个实际段落（EXT，一个实际段落占 16K 字节），文件存取属性是读写型。

又如:

A>STAT * .BAS↓

屏幕显示:

RECS BYTES EXT ACC

14　2k　1　R/W　A:ABC.BAS

20　4K　1　R/W　A:DEF.BAS

48　6K　1　R/W　A:HIJ.BAS

BYTES REMAINTING ON A:100k

表明驱动器 A:中磁盘扩展名为 BAS 的文件有 3 个，并列出了这三个文件各占的记录数、长度、实际占用段落数、属性及文件名，磁盘的可用空间等。

4. 指定文件属性

格式：STAT　X:〈文件名〉$〈属性〉

作用：为文件指定属性。

CP／M 系统规定具有下列文件属性：

R／O，只读属性，具有这种属性的文件只能读入，不能改变和删除。

R／W，读写属性，具有这种属性的文件可读、可改、可删。一般未指定属性的文件都是 R／W 型的属性。

SYS，系统文件属性，在 DIR 命令文件目录中不出现具有这种属性的文件名。可以修改、删除或读出具有这种属性的文件。STAT *.* 命令的目录中可以看到属性为 SYS 的文件。

DIR，在 DIR 命令的目录中能显示的文件。

指定文件属性实例：

①对文件指定只读属性

A>STAT　ABC.BAS $ R／O ↓

屏幕显示：

ABS.BAS SET TO R／O

表明文件 ABC.BAS 已被指定为只读文件。

②对文件指定系统属性

A>STAT　FILE.BAS $ SYS ↓

屏幕显示：

FILE.BAS SET TO SYS

表明指定文件 FILE.BAS 为系统文件，用 DIR 命令列目录看不见这个文件名。

5. 显示磁盘特性

格式：STAT DSK: ↓

作用：显示驱动器中磁盘特性。

例如：

A>STAT DSK:↓

A:Drive characteristics

1024: 128 Byte Record Capacity

128: Kilobyte Drive Capacity

48: 32 Byte Directory Entries

48: Checked Directory Entries

128: Records / Extent

8: Records / Biock

32: Sectors / Track

3: Reserved Tracks

上述特性是：磁盘共有的记录数；总容量；最多可存储文件数（48）；每个目录入口至多可有记录数（128）；文件定位至少要求磁盘空间的记录数（8）；每个磁道分成的扇区数（32）；不能存文件的磁道数（3），这些磁盘特性与所用的CP / M系统有关。

6. 显示外部设备配置情况

格式：STAT DEV:↓

作用：显示微机外部设备配置情况。

例如：

A>STAT DEV:

CON:is CRT:

RDR:is PTR:

PUN:is PTR:

PST:is LPT:

上面的显示中，左边代表逻辑装置，右边代表指定给逻

辑装置的外部设备。在 CP／M 中有 4 种逻辑装置，即：

CON:　接收命令，显示信息，控制台。

RDR:　接收信息，读入纸带。

PUN:　输出信息，纸带凿孔。

LST:　输出信息，打印列表。

配合这 4 种逻辑装置的是 12 种外部设备：

TTY:　慢速主控（电传打字）机。

CRT:　快速主控（CRT 显示器）。

BAT:　批量处理器。

UCT:　用户定义的主控机。

PTR:　纸带读入机。

PTP:　纸带凿孔机。

UR1:　用户读入机#1。

UR2:　用户读入机#2。

UP1:　用户凿孔机#1。

UP2:　用户凿孔机#2。

LPT:　行式打印机。

UL1:　用户列表装置。

这种配置方式对于指定的 CP／M 系统，厂家已经指定，若有必要，用户可以用 STAT 命令改变配置情况。

7. 改变微机外部设备配置的命令

格式：STAT　log:＝phy

作用：改变微机外部设备配置情况。

用这个命令指定实际外部设备（phy:）至逻辑装罝（log:），在一个命令行中可以指定几个外部设备的改变情况，中间用逗号隔开。

例如：

STAT　log:=phy:, log:=phy:　↓

8.　显示设备分配情况及 STAT 命令使用情况

格式：STAT　VAL:↓

作用：显示所有设备分配及 STAT 命令使用情况。例如：

A>STAT　VAL:↓

Temp R/D Disk:d:=R/O

set Indicator: d: filename.typ $ R/O $ R/W $ sys $ DIR

Disk Status: DSK: d: DSK:

USer Status: USR:

Iobyte Assign:

CON:=TTY:CRT:BAT:UC1:

RDR:=TTY:PTR:UR1:UR2:

PUN:=TTY:PIP:UP1:UP2:

LST:=TTY:CRT:LPT:UL1:

上面所列是 STAT 命令用法的简单提示，第 1 行表示键入 STAT d:=R/O，可以使驱动器 d:中磁盘为暂时只读型磁盘；第 2 行表示如何指定文件的属性；第 3 行说明如何显示磁盘特性；第 4 行表明如何显示用户及正在操作的磁盘文件号；第 5 行表明如何指定实际外部设备至逻辑设备。

9.4.5　外部命令的出错信息

1.　BDOS ERR ON d: BAD SECTOR

可能是磁盘没有格式化，或磁盘上某一扇区有问题，或磁盘没能放好。

2.　BDOS ERR ON d: SELECT

驱动器指定错或门没关好，或电源未打开。

3. BDOS ERR ON d: R/O

磁盘写保护，或换了磁盘没有执行热引导。

4. PIP 命令错误

DISK READ ERROR	读磁盘错。
DISK WRITE ERROR	写磁盘错。
VERIFY ERROR	常与 BDOS 错同时发生。
NOT A CHARACTER SINK	不能把字符送到该处。
READER STOPPED	读入设备停止。
NOT A CHARACTER SOURCE	不能取得字符。
ABORTED	处理过程终止。
BAD PARAMETER ()	参数错。
INVALID USER NUMBER	用户编号错。
RECORD TOO LONG	记录太长。
INVALID DIGIT	格式不符。
END OF FILE CTL-Z?	文件末尾，要求 CTRL-Z 键予以肯定。
CHECKSUM ERROR	Intel 格式与读入记录不符。
CORRECT ERROR	校正错。
INVALID FORMAT	格式错。
NO DIRECTORY SPACE	文件目录错。
NO FILE	不存在指定的文件。
START NOT FOUND	不能找到起始字符串。
QUIT NOT FOUND	不能找到终止字符串。
CANNOT CLOSE FILE	不能关闭文件。
DESTINATION IS R/O	目的文件为只读型，若按 Y 键删除文件后复制文件，若

按 N 不删除，中止复制。

UNRECOGNIZED DESTINATION　指定的外设不存在。

CAN NOT WRITE　不能写。

INVALID PIP FORMAT　命令格式错。

CAN NOT READ　不能读。

INVALID SEPARATOR　标点错。

NOT FOUND／NOFILE　没找到／没文件。

REQUIRES CP／M2.0　PIP 与 CP／M 不匹配错。

5.STAT 命令出错

Bad Delimiter　定界符位置错。

Invalid Assignment　指定设备格式错。

Invalid File Indicator　指定文件格式错。

* * TOO MANY FILES * *　超过 STAT 能重排的文件数量。

Invalid Disk Assignment 指定磁盘错。

Wrong CP／M Version　STAT 与 CP／M 不匹配错。

习题九：

1. 上机实习本章所介绍的操作系统命令。

附录　命令索引（按字母排序）

为了使计算机知识早日在我国得到普及，我部应用户要求，特邀请四川大学计算机系韩仲清，廖兴祥老师在百忙中为我们编写了这本书，意在让广大的用户和计算机爱好者能够更早的了解和使用中华学习机。我部自经营中华学习机以来，得到了广大用户的大力支持和信赖。在此，谨借本书致谢。今后如果你在学习和使用中华学习机时遇到困难，三开元是你最好的朋友。我们备有中华学习机上所用的工具软件、教学软件、游戏软件、游戏磁带数百种及齐全的外围设备，随时供你选用，并免费提供清单。函索必寄。

中华学习机汉字数据库的应用

成都三开元电脑部